Computing today

OCTOBER 1982

75p

Financial planning for all: DIY spreadsheet program

WIN A SPECTRUM SYSTEM

Speed up your programming

System simulation on a micro

Sharp Practice? User report on the MZ-80A

CONTENTS

VOL 4 NO 8 OCTOBER 1982

EDITORIAL & ADVERTISEMENT OFFICE
145 Charing Cross Road, London WC2H 0EE.
Telephone 01-437 1002-7. Telex 8811896.

Editor :
Henry Budgett
Advertisement Manager:
Neil Johnson
Editorial Assistant:
Roger Munford
Advertisement Copy Control :
Sue Couchman, Sonia Hunt
Managing Editor :
Ron Harris BSc
Managing Director :
T J Connell

Origination and design by MM Design & Print.

ABC Member of the Audit Bureau of Circulation

ISSN 0142-7210

Computing Today is normally published on the second Friday in the month preceding cover date. Distributed by: Argus Press Sales & Distribution Ltd, 12-18 Paul Street, London EC2A 4JS. 01-247 8233. Printed by: Alabaster Passmore & Sons Ltd, Maidstone, Kent.

Subscription Rates : UK £12.10 including postage. Airmail and other rates upon application to Computing Today Subscriptions Department, 513 London Rd, Thornton Heath, Surrey CR4 6AR.

Computing Today is constantly on the look-out for well written articles and programs. If you think that your efforts meet our standards, please feel free to submit your work to us for consideration.

All material should be typed. Any programs submitted must be listed (cassette tapes and discs will not be accepted) and should be accompanied by sufficient documentation to enable their implementation. Please enclose an SAE if you want your manuscript returned, all submissions will be acknowledged. Any published work will be paid for.

All work for consideration should be sent to the Editor at our Charing Cross Road address.

CONSUMER NEWS

£9 MILLION FOR SCHOOLS

Last year, the Government allocated a substantial grant to secondary schools allowing them to acquire at least one micro and familiarise themselves with computer science. This year, it is the turn of the primary schools.

Primary schools will be able to apply to their local education authorities for one of the three Government approved micro packages: the Sinclair ZX Spectrum (48K), the BBC Model B complete with disc interface, and the Research Machines Link 480Z. The Government expect around 2,700 schools to apply for computers and have thus allocated £9 million to fund the project.

A hope for the future might be that this token acknowledgement of computer science will cause enough interest to encourage the Government to provide more than just ***one*** micro to each school.

BUG BYTES

Two howlers of a somewhat gross nature this month. First, the Reflections article on 3-D graphics by Jack Dikian and Damien Skracic. The program listing appears to have a number of rather fundamental bugs in it . . .to put it mildly! However, for once the blame for these cannot be laid totally at our door as the original listing they supplied also contains the bugs. We are currently awaiting a reply from the authors,they live down under! As soon as these arrive corrections will be published.

The second appearance of the gremlin came in the FORTH Simulator published in the June issue. Blame for the mistakes can safely be attributed to the Editor who claims that a heavy Saturday night blurred his eyesight for the Sunday morning typesetting session . . .personally, I don't believe a word of it! The major errors are as follows:

```
  20  CLEAR 2500:DEFSTR A,B,S:
      DEFINT C-R,T-Z:DIM A(40):
      DIM AC(20,1):DIM AV(20,1):
      DIM AW(20,10):NW=20:
      DIM N(20)
  80  FOR J=0 TO 40:A(J)="":NEXT:
      T=0:C=0:D=0
1070  IF A(J)="I" THEN
      AS=STR$(DS):GOSUB 3000
1200  IF A(J)="-" THEN
      AS=STR$(VAL(S(1))-VAL(S(0))):
      GOSUB 4000
1220  IF A(J)="/" THEN
      AS=STR$(INT(VAL(S(1))/
      VAL(S(0)))):GOSUB 4000
```

A number of people commented that the program did not offer all the facilities that they would expect of FORTH. The only answer to this is that they are quite right, it didn't . . .but then again it was only supposed to give an idea of the way FORTH works. If you found that you got on well with it and wanted to do more, then you have just proved to yourself that what you really need is a FORTH interpreter!

THE CLASSROOM COLLECTION ▼

With news of the Spectrum about to become the blackboard of the '80s, here's some good news for all you ZX81 owners. Griffen & George have substantially expanded their range of accessories to include a desk console and three interface units.

The desk console, designated CRA-714-N and priced at £33.00 keeps ZX equipment (ZX81, 16K RAM pack, ZX printer, PSU, cassette recorder, cassettes, etc) together, secure and stable.

The three interface units comprise: a RAM I/O pack, CRA-720-H, which contains an additional 4K RAM; a Control pack, CRA-540-U, which contains eight relays each capable of switching 1A at up to 30V; and lastly, the Analogue pack, CRA-724-520D, which will convert an analogue input voltage in the range 0-2.55V to a digital signal to feed the computer. The RAM I/O pack, Analogue pack and Control pack are priced at £35.54, £18.69 and £28.05 respectively.

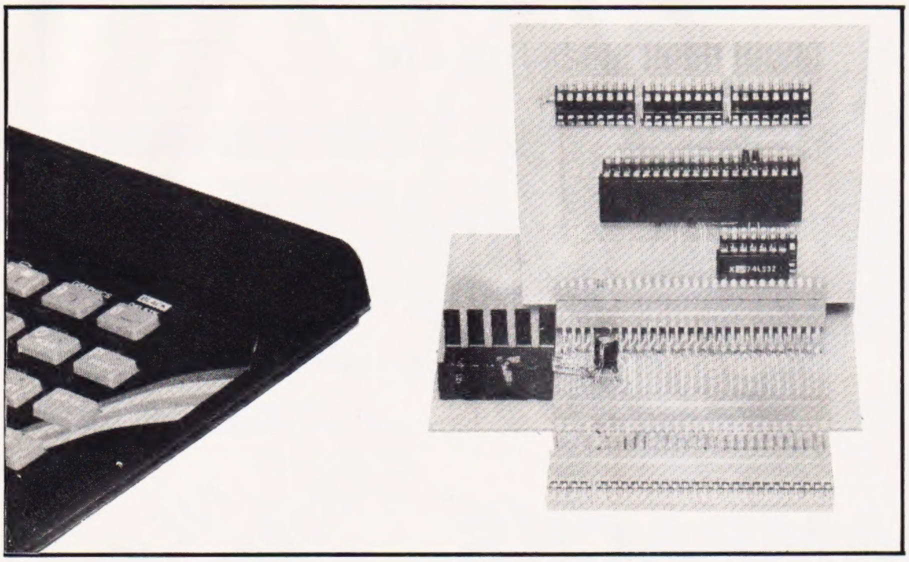
For more details on these and their other educational products contact Griffen & George Ltd, 285 Ealing Road, Alperton Road, Wembley, Middlesex HA0 1HJ or telephone 01-997 3344.

SOUNDS GOOD

If you thought your 'Space Invaders' programs lacked that certain something, maybe you ought to check out the ZON X-81 Sound Unit.

Completely self-contained in a black plastic case with loudspeaker and manual volume control, the unit plugs in between the rear of the ZX81 and its RAM pack and/or printer.

The ZON X-81 is based on a three-channel-plus-noise chip and allows the pitches and volumes of the three channels and the overall attack/decay envelope to be controlled by simple BASIC statements. In this way, you can simulate the sound of piano, organ, bells, helicopters, lasers and explosions — the stuff of which good graphics games are made!



Priced at £25.95, you can find out more about the ZON X-81 from BI-PAK Semiconductors, PO Box 6, Ware, Herts or you could always telephone them on 0920-3442.

SPECTRUM'S MOTHER? ▲

It never fails! Clive Sinclair introduces a new ZX product and pretty soon the whole market is flooded with add-ons. So, it came as no surprise to hear news of a motherboard and a 24 line I/O port custom designed for the ZX Spectrum.

The device includes three eight-bit I/O ports (marked A,B and C) which can be written to or read from. The unit is memory mapped and access to it can be made by one simple BASIC command. Available fully built and tested, complete with instructions and control applications, the port can be used either with the motherboard (thus allowing a further card to be used) or with a stackable connector.

The ZX Spectrum user I/O port, ZX 2 slot motherboard and stackable connector are priced at £16.50, £16.95 and £5.50 respectively. All the prices are inclusive of VAT but you'll have to add 70 pence if you want postal delivery.

For more details on these devices contact Kempston (Micro) Electronics, 60 Adamson Court, Hillgrounds Road, Kempston, Bedford MK42 8QZ.

SOME LIKE IT HOT . . .

And if you're one of them, you may be interested to hear about the K160, a 16-column miniature thermal printer mounted on a printed circuit board.

The unit consists of a print head, the SP285, a PSU, drive electronics and an interface to an eight-bit parallel bus which allows simple connection to a microprocessor system. ASCII data is sent to the K160's customised microprocessor which controls all the timing, drive and character generating functions of the print head.

The printer uses 38mm wide thermal paper and prints at 1.75 lines per second. To ensure efficient operation, the K160 incorporates programmable line feed and self test programs.

Priced at £87.10, you can get more information from Roxburgh Printers Ltd, 22 Winchelsea Road, Rye, East Sussex or you can 'phone them on 079-73 3777.

THE FOUR TOPS ▼

One quick rub on the magic lamp and Lowe Electronics have introduced a new computer to the marketplace – the Genie IV Colorcomputer EG2000.

Based on the Z80, the Genie IV comes complete with 16K RAM (internally expandable to 32K), 16K Extended BASIC, serial and parallel I/O, and the provision for 12K of plug-in program cartridges. There is also a facility for a light pen.

The keyboard is typewriter style with four programmable keys. The Genie IV also boasts a 40 by 25 screen format with a full graphic resolution of 160 by 80 pixels and is able to reproduce 128 graphic characters plus a further 128 user programmable graphic character patterns. Nine colours are available for character display and four colours for full graphics display.

The asking price for the Genie IV is £199.99 but for the full facts and figures, have a quick word with the people at Lowe Electronics, Chesterfield Road, Matlock, Derbyshire DE4 5LE. Telephone enquiries can be made on 0629-4995.

You might also like to ask them about the LE18 HI-RES unit for the Genie I and II systems. Priced at £86 plus VAT, the unit offers bit image graphics of 73,728 points (a resolution of 384 by 192); this is achieved using a separate 16K of video memory.

Its graphics can be fully intermixed with text or existing pixel graphics and a custom designed extension to the Genie BASIC interpreter provides commands including PLOT, VECT, FILL, REV, CLR, DISP, HOLD and VIEW. The LE18 HI-RES unit also includes a scratchpad memory allowing animation and the use of programmable graphics characters.

BUSINESS NEWS

CP/M ADD-ON

CP/M is now available as an add-on option for the Cluster/One personal computer network.

Cluster/One CP/M has the following features: eight different volumes can be accessed simultaneously; variable sized disc from 32K to 8M can be defined on hard disc; it can support both Apple minifloppy disc and IBM-standard single density discs; and it can support 56K of RAM per user station. This addition to the Cluster/One network also allows user access to a wide range of applications software including Wordstar and dBase II.

CP/M is additionally supported on the network by an optional add-on feature of the existing Network File Server which provides shared storage for up to 64 Apple II and Apple III workstations. Each Apple II user station running CP/M requires a Microsoft SoftCard.

For more details contact Zynar Ltd, 122/123 High Street, Uxbridge, Middlesex UB8 1JT or telephone them on 0895-59831.

A MIMI MICRO?▼

A business and professional microcomputer, designated the MIMI 802, has been developed in conjunction with the Design Council's Design Advisory Service.

Based on the Z80A, the 802 incorporates integral double sided double density floppy disc drives providing 700K of memory. Standard interfaces include full RS 232C, Centronics parallel and a light pen facility for interactive programming. The machine's keyboard has a standard QWERTY format with 17 function keys and allows individual configuration of other character sets from Arabic to scientific characters. The screen format of the 802 is 80 by 25 with high resolution graphics on the 802G (the graphics version of 512 by 256 pixels

The MIMI's operating system, OS/M, is CP/M compatible and there is also a utility for the conversion of SuperBrain discs to MIMI format. A number of application packages are also available to run on the machine.

The MIMI 802 is priced at £1,350 with the 802G at £1,495. For further details get in touch with British Micro, Penfold Works, Imperial Way, Watford, Herts WD2 4YY or 'phone 0923-48222. You might also like to ask them about the Winchester disc option they are developing for this system.

WHEELS WITHIN WHEELS

An upgraded version of the Daisywriter printer 1000 has been developed — called the Daisywriter 2000.

With the ability to be connected to any of the usual serial or parallel interfaces, the Daisywriter 2000 has a 16K or 48K buffer memory. The printer can also be used with all popular word processing packages including Wordstar, Applewriter, Spellbinder, Easywriter, etc

Enclosed in dust-free cassettes, the range of Daisywriter print wheels include over 100 different character sets and typestyles covering various language alphabets and they have a minimum life of 10 million impressions.

Priced at £995 plus VAT Daisywriter 2000 is available from any of CPU's network of dealers in the UK. For more detailed information contact CPU Peripherals Ltd, Rodd Industrial Estate, Govett Avenue, Shepperton, Middlesex TW17 8AQ or telephone Shepperton 46433.

WHERE EAGLES DARE . . .

Two microcomputers have been added to the Eagle Business System range: the Eagle IV and the Eagle V.

Formatted built-in hard disc capacity of the Eagle IV and the Eagle V is 8.284M and 15.784M respectively. This capacity can be added to with additional hard discs which can be plugged into the internal controller or to the system interface. The internal controllers in these systems can drive up to two Winchester discs and feature an 11-bit error correction capability.

The Eagles are supplied with a menu program which is automatically loaded on power up and the system comes complete with at least £600 of software including BASIC, Ultracalc (a spreadsheet program), CP/M operating system and AVL's Spellbinder word processor with mail merge and mail list management.

The price of the above starts at around £4,195. However, there is an alternative software package with all of the above plus a seven ledger integrated accounting

program called Accounting Plus. The cost of this second option starts at around £5,400 which includes over £3,000 worth of software.

Further information is obtainable from Mediatech, Business Systems Division, Woodside Place, Alperton, Wembley HA0 1XA or by telephoning 01-903 4373.

BEST OF THREE ▼

The Genie III is a complete microcomputer business package aimed at small firms and the professional market. The package, priced at £3,250 plus VAT, comprises a microcomputer, printer, relevant software programs and a one year guarantee contract covering hardware and software.

The Genie III itself is a fully expanded and integrated micro with 64K RAM, a built-in screen, dual disc drive with a double density storage facility and a full size keyboard with numeric keys. The printer is the EG 3085 matrix model with bi-directional movement and a print speed of 100 characters per second.

The software included in the deal is worth nearly £700 and is written and supplied by TABS. It covers sales ledger, purchase ledger and nominal ledger systems.

Initially, the Genie III package will be available only through 12 'superdealers' but expansion of the dealer network is expected within a year. For further information get in touch with Lowe Electronics, Chesterfield Road, Matlock, Derbyshire DE4 5LE or ring on 0629-4995.

THE ACTING IBM?

MSDOS, the Microsoft operating system commissioned by IBM for use on their Personal Computer, is now available to run on the ACT Sirius 1.

ACT are also introducing several languages to complement the MSDOS including compiled and interpreted BASICs by Microsoft, Pascal, FORTRAN and COBOL as well as an MSDOS Assembler called Macro 86.

MSDOS is being included with the Sirius system at no extra cost, which is quite generous as the Sirius already comes complete with the CP/M 86 operating system. The ACT Sirius 1 is priced at £2,395 plus VAT and comes with swivel and tilt screen, separate keyboard and 1.2M integral floppy discs.

For more information about the IBM-compatible MSDOS, get in touch with ACT (Sirius) Ltd, Shenstone House, Dudley Road, Halesowen, West Midlands or 'phone 021-501 2284. And if you're wondering why the ACT personnel sound so happy on the telephone it could be because they know the IBM Personal Computer still won't be on these shores for a few months yet!

ON YOUR DESK

Two versions of the Z100 series of desktop computers are now available for the business, professional and OEM markets.

Featuring as standard both an eight-bit (8085) and a 16-bit (8088) microprocessor, the systems incorporate a five slot S100 expansion chassis, the choice of either two built-in 320K 5¼" floppy disc drives or a 5M Winchester drive plus floppy back-up, 128K RAM expandable up to 786K and high resolution graphics (eight colours) with a pixel resolution of 144,000 dots.

Two microcomputers will be available in the range: the Z-100, a 'low profile' computer with no monitor; and the Z-120, an 'all in one' version with a built-in green monochrome video display. Both machines have white and brown housings and occupy a desktop area of less than 20" by 20".

For further information, including full pricing details, write to Zenith Data Systems, Heath Electronics (UK) Ltd, Gloucester GL2 6EE.

DISCOUNT DEAL

Triumph Adler, confident with high volume sales of their microcomputer range, have announced a number of price reductions to their Alphatronic microcomputer series.

The P2/48K Alphatronic, complete with VDU and twin single sided double density floppy disc drives, has been reduced from £2,095 to £1,895. The P2/64K version (with all the above features) has fallen from £2,295 to £2,095 and the model P2/U, with 64K of RAM and twin double sided double density floppy disc drives is down from £2,495 to £2,295. They have also reduced the price of the DRH 80 bi-directional needle matrix printer from £560 to £450.

In addition, Triumph Adler have introduced an IEC Bus enabling users to link their micro to a wide range of IEEE-based peripherals and instrumentation. The price of this interface is £150.

For further information contact Triumph Adler UK Ltd, 27 Goswell Road, London EC1M 7AJ. Telephone enquiries can be made on 01-250 1717. ■

Probably the fastest microcomputer in the universe

the JUPITER ACE only £89.95.

All inclusive Price

For £89.95 you receive your Jupiter Ace, a mains adaptor, all the leads needed to connect to most cassette recorders and T.V.s (colour or black and white), a software catalogue and a manual.

The manual is a complete introduction to the world of personal computing and a course in FORTH programming on the Ace.

Even if you are a complete newcomer to computers, the manual will guide you step by step from first principles to confident programming.

The price includes postage packing and V.A.T.

Key Features

- Revolutionary microcomputer language FORTH.
- Full-size moving-key keyboard.
- User-defined high-resolution graphics.
- Programmable sound generator.
- Floating point arithmetic.
- Fast cassette interface.
- Upper and lower case ascii character set.
- 24 x 32 character flicker-free display.

The Jupiter Ace uses FORTH

The Ace is set apart from all other personal computers on the market by its use of a revolutionary language called 'FORTH'. Some computer languages are easy for humans to understand, others are easy for computers; FORTH is most unusual in being both. Its underlying principles are so simple that it takes even a newcomer to computers only a few minutes to learn how to do calculations on the Ace, yet the very same principles are powerful enough to allow you to invent your own extensions to the language itself.

At the same time, the memory-saving coded form used to store your programs inside the Ace allows it to obey them very fast — typically in less than a tenth of the time it would take to do the same thing using a different language. Amongst other things, this makes the Ace ideal for games.

FORTH's unique combination of speed, versatility and ease of programming has already made it a prime choice for professional applications as diverse as pub games and radio telescopes, and gained it an enthusiastic national user group. Now the Jupiter Ace can bring this addictive language into your own home.

Designed by Jupiter Cantab

Leading computer Designers Richard Altwasser and Steven Vickers have a reputation for pushing technology forwards. After playing the major role in creating the ZX Spectrum they formed Jupiter Cantab to develop their latest brainchild the Jupiter Ace.

Technical Specification

Hardware

Processor/Memory
Z80A running at 3.25 MHz. 8K bytes ROM 3K bytes RAM.

Input
40 moving-key keyboard with auto-repeat on every key.

Output
Memory-mapped 32 x 24 character display with high resolution user graphics. Output to drive normal UHF TV set on channel 36.

Sound
Provided by internal loudspeaker.

Cassette
Load Save & Verify at 1500 baud, separate data storage.

Software, FORTH

Data Structures
Integer, Floating point and String data may be held as constants, variables or arrays with multiple dimensions and mixed data types.

Control Stuctures
IF-THEN-ELSE, DO-LOOP, BEGIN-WHILE-REPEAT, BEGIN-UNTIL, all may be mixed and nested to any depth.

Operators
Mathematical +, —, X, ÷.
Logical AND, OR, NOT, XOR.
Comparison <, >, =.

Program Editing
FORTH words may be listed, edited and redefined. Comments are preserved when words are compiled.

Order Form

The **Jupiter Ace** is available only by mail order. Please allow up to 28 days for delivery.

Send cheque or postal order with the form to:—
JUPITER CANTAB, 22 FOXHOLLOW, BAR HILL, CAMBRIDGE CB3 8EP

Please send me:—

☐ JUPITER ACE MICROCOMPUTER(S) @ £89.95.

Name. Mr/Mrs/Miss ____________________

Address ____________________

____________________ D

SOFT WARES

HOW ARE YOU MANAGING?

An assembly language relational data base management system, dBase II, is available for use on a 64K Apple II with either a Microsoft SoftCard or an ALS Z Card installed.

Using the package, users can produce custom designed information management systems of varying complexity. Most dBase II commands are verbs supplemented with programming structures; these commands sound very English-like thus making it easier for non-programmers to utilise the system.

Applications for the dBase II disc include general ledger, accounts receivable, accounts payable, stock control, payroll and job cashing.

The cost of the dBase II package is £299 and it is available on a two week sale-or-return basis for evaluation. You can find out more about dBase II from Pete & Pam Computers, Waingate Lodge, Waingate Close, Rossendale, Lancs BB4 7SG or by telephoning 0706-227011.

DO YOU LISP?

LISP, the list processing language, is now availabe in cassette form for the ATOM and BBC Micro priced at £17.25 and £16.95 respectively. Developed by Owl Computers in conjunction with Acornsoft, the implementations are accompanied by specially written manuals which serve as an introduction to the language.

There are also two books, 'LISP Theory and Practice' and 'LISP on the BBC Microcomputer', available for £6 and £7.50 respectively.

The cassettes and books can be obtained from Acornsoft via mail order or through registered Acorn dealers. For further information contact Acornsoft at 4a Market Hill, Cambridge CB2 3NJ.

THE SEARCH IS ON

Since founding their company, Bug Byte, way back in 1980, Tom Baden and Tony Milner have sold more than half a million games cassettes for micros.

Bug Byte's cassette range, now 30 strong, is available at present through 90 dealers nationwide. However, Bug Byte are looking to increase the size of their dealership to 500 within the next 12 months — no small task! And that's where you come in.

As an incentive, Bug Byte are going to be offering cash purchase discounts, special monthly prizes, point-of-sale material, promotional packs as well as a 12 month guarantee to all dealers. So, if you are interested in becoming a Bug Byte dealer or just want to find out a few more details of their range of software for the Commodore VIC, ATOM, BBC Micro and Sinclair systems (to name but a few), why not get in touch with Bug Byte at 98 The Albany, Old Hall Street, Liverpool L3 9EP or 'phone them on 051-227 2642.

PLAN AHEAD ▼

Why is this man smiling? Perhaps it's because Comshare have just released a financial modelling system called PLANNERCALC which is expected to retail for a mere £39.

Employing the 'spreadsheet' approach, the user can enter new figures or rules, immediately viewing the effect on the data already in the model. PLANNERCALC also incorporates mainframe features — using PLANNERCALC, the user does not have to number the model's rows in the correct logical sequence as the rows will be sorted as the calculations are made.

For users requiring additional facilities such as consolidation, an interface with word processing systems, data bases, and other systems, as well as enhanced reporting, a more sophisticated modelling system is available called MASTERPLANNER. (If you wish to upgrade your system to MASTERPLANNER at a later stage, Comshare will let you trade in your old PLANNERCALC.)

Both PLANNERCALC and MASTERPLANNER will run on most micros which operate under CP/M, have 64K memory, an 80 column screen and 5¼" or 8" disc drives, including the SuperBrain, the enhanced Apple II, TRS-80, ACT Sirius 1 and the IBM Personal Computer (when it arrives) and a whole host of others.

Complete details of both PLANNERCALC and MASTERPLANNER are available from Comshare Ltd, 32/34 Great Peter Street, London SW1P 2DB or by telephoning 01-222 5665. And, if the man in the photo seems familiar, think no further back than the BBC's Computer Programme — Ian McNaught-Davis (Comshare's managing director) was one of the presenters. ■

NEXT MONTH

computing today

NOVEMBER ISSUE
ON SALE
FRIDAY 8th OCTOBER

TESTING DAYS

Once you've produced that elegant piece of software that's taken up so many late evenings and stressed your personal relationships almost to the brink of the Divorce Court, you may well be thinking of sending it to a commercial software house or magazine. Er . . . don't. Or rather, not yet! One of the first tests made on a piece of commercial software by the bands of hardened and cynical reviewers is to give it an impossible date. If the program blindly accepts the fact that the 32nd of February really exists, that's about as far as the program ever gets!

In our next issue we'll be looking at the ways in which you can test your software to destruction before anyone else gets near it, and we'll be giving one way of sorting dates out too, just to round things off.

WHAT'S IN A NAME?

First there was the PET and then they changed its name. Then came the SuperPET and they gave it a number. Now there is the Commodore 64 and . . . well, it'd spoil the fun if we gave away all the secrets about the new system from the Big C, wouldn't it. So, if you want to find out what 64K of RAM and colour and sprites and a whole host of special features can do for your life, then don't miss next month's issue of the magazine. After all, if you miss this, you'll also miss the second instalment of your free book too and you wouldn't want to do that . . . would you?

MAKING MUSIC

A well-known playright once wrote "If music be the food of love . . . play on" and ever since the staple ingredient of restaurants has been Musak! Just what *is* music and how can computers produce it, after all one is analogue in nature and the other digital? In next month's issue we take a long hard look at the subject of computer music and how to go about writing your own composer as well as looking at the ultimate musical instrument, a mere £18,000's worth of computerised Fairlight.

So, whether you like Beethoven, Borge or Bad Company or are just a 'bit' interested, tune into our November issue . . . .

COMMODORE COMMUNICATIONS

If you've ever wished that you could pass programs between computers just by adding a piece of wire now is your chance to make that dream come true. Two simple routines and a length of ten-way cable are all that is needed to join any New ROM Commodore machine to either another of the same type or a VIC system. Programs can now be passed at speeds in excess of those achieved on disc making this an ideal utility for those involved in education or software development.

GIVE THIS TO YOUR NEWSAGENT
Please reserve me a regular copy of Computing Today

NAME ....

ADDRESS ....

POSTCODE ....

Trevor Hamilton-Gibbs

SYSTEMS SIMULATION

Forward planning for a business with the help of a TRS-80.

Some while ago I took on the responsibility to provide a microcomputer-based system for my office to carry out a large number of relatively simple but highly repetitive calculations. This system was to be the forerunner of a much larger mainframe-based system. Initially I selected a TRS-80 with discs, printer, and the full 48K memory, which was ideally suited to the task as all the basic information was at that time held in an off-site system to which we had only manual access. The actual task of collating data for input was more than outweighed by the manipulative convenience once in the micro. I originally estimated that the payback period on the investment would be around nine months; in fact, due to the amount of work pushed through the machine the payback period was reduced to around four months.

That is all very interesting, but what has it got to do with the reason for this article — not a lot, it is true, except that some of the future development could be simulated very easily on the TRS-80. The micro was installed some seven months before a number of IBM 3278 VDUs were introduced to the staff. During that seven months it was decided that simulation of as many of the 3278 facilities as possible would provide a very good basis for the training of the users, not just in 3278 operation but also to understand more of the results that were provided by the calculations. It is just one these facilities which forms the rest of this article.

Of course, only limited simulation could be given in respect of the full range of special functions contained in such an advanced terminal as a 3278 but full cursor control could be effected very easily. As I had already specified the major needs of the on-line systems being accessed by the IBM 3278, it was an easy matter to transfer parts of this for the simulation.

The system was designed to be menu driven using cursor control, null entry and programmed function keys rather than an alphanumeric character input procedure. While it is seemingly just as fast to use the latter on a mainframe, it can be much slower on a micro and uses more memory and as that is usually at a premium, the fewer the program statements to drive the system the better.

WHAT'S ON THE MENU

Listing 1 (Cursor Control Menu Demonstration) gives a very simple and short program which sets up a menu display of five facilities and some instructions and then positions a cursor character to the left of the first entry in the list. The user can then move this cursor up or down, but not beyond the top or bottom of the list of facilities.

For those readers among you who are not familiar with some of the TRS-80 Disc BASIC commands and characters, a short explanation should make this listing a lot clearer.

```
10  CLEAR 500
20  A=14400:U=8:D=16:E=1:TL=201:LL=713:MO=128
30  DEFUSR0=&HFFF2
40  ME$=CHR$(10)+CHR$(208)+"CURSOR CONTROL MENU
    DEMONSTRATION"+CHR$(10)
50  FOR I=1 TO 5:ME$=ME$+CHR$(10)+CHR$(205)+"FACILITY
    NUMBER "+CHR$(I+48)+CHR$(10):NEXT
60  ME$=ME$+CHR$(10)+CHR$(10)+CHR$(197)+"POSITION
    CURSOR AT REQUIRED FACILITY AND PRESS 'ENTER'"
70  C$=CHR$(140):D$=CHR$(32)
80  FOR I=(&HFFF2) TO (&HFFFF):READ M:POKE I,M:NEXT
90  CLS:PRINT ME$:CP=201
100 PRINT@CP,C$;
110 IF PEEK(A) AND D AND CP<LL THEN PRINT@CP,D$;:
    CP=CP+MO:GOTO 100
120 IF PEEK(A) AND U AND CP>TL THEN PRINT@CP,D$;:
    CP=CP-MO:GOTO 100
130 IF PEEK(A) AND E THEN 140 ELSE 110
140 POKE (&HFFF6),(150+(CP-137)/128*8)
150 X=USR0(0)
160 FOR I=1 T0 1000:NEXT:GOTO 90
170 DATA 33,0,60,54,0,17,1,60,1,255,3,237,176,201
```

Listing 1.

Line Function

Lines 10-20: CLEAR 500 ensures that sufficient string space is reserved for the three strings used in this demonstration. Variable A is assigned the decimal number of an I/O address, more of which later. Variables U,D and E are assigned the three possible decimal values which can occur in this demonstration in address A. TL is assigned the screen memory map limit of the topmost cursor position and LL is assigned the lower limit. MO is assigned the number of screen print positions I want to move the cursor each time.

Lines 30, 80 and 140 — 170: The DEFUSR statement assigns the entry point in hexadecimal of a short machine code routine. This routine has nothing to do with cursor control, it is just my way of demonstrating that you are beyond the menu facility selection point. All the routine does is to zap up on the screen 1024 identical characters. For BASIC II users line 30 will have to be changed to:

```
POKE 16526,242:POKE 16527,252
```

but, of course, you can delete line 80 and replace lines 140 onward by your own program. The machine code entry point is for a 48K system; you will have to re-calculate this if you have less memory or already have some other machine code routine stored at the top of memory, such as Renumber, Kbfix or Clock. You must also reserve space when replying to the memory size request as BASIC is initialised; this routine needs 14 bytes. The actual machine code values (in decimal) are held in DATA statements and written away to high memory in line 80. The routine requires one extra value and this is POKEd to the correct address in line 140. The actual value POKEd is calculated from the cursor position and the formula used ensures that the zap gives a different display for each of the five facilities if Enter is pressed. Line 150 is the entry to the machine code routine and line 160 is a short delay loop to leave the display on the screen before re-displaying the menu.

Lines 40 to 60: This rather complicated series of assign statements actually creates one string, ME$, which prints the whole menu onto the screen in one go—it is very fast. The string contains a number of special characters — CHR$(10) is the code to perform a Line Feed and returns the cursor to the left most position on that line and CHR$ (197), (205) and (208) are space compression characters which work the same as TAB statements but inside a string. CHR$(I + 48) creates the string representation of the number in I each time

the loop is gone through and CHR$(48) is the string character zero. (Page 93 of *Computing Today*, Vol. 3 Number 9, November 1981, has a full list of TRS-80 character values.) The full ME$ can be seen set out as it would be displayed in Fig. 1.
Line 70: C$ is the cursor character; in this case CHR$(140) (see the pixel codes on the CT Standards page, it is actually P12), but any suitable cursor character can be used. D$ is just a blank space used to remove the old cursor as it is moved to a new position.
Line 90: CLS blanks the screen out completely removing all characters. CP is assigned 201, which is the initial cursor position, 10 spaces in from the left side of the screen and four lines down, just to the left of facility number one.
Lines 110 to 130: These three lines contain all statements which are needed to move the cursor character up and down and detect whether the Enter key has been pressed. Address 14400, now accessed by PEEK(A), is a keyboard I/O address which is common to the cursor control keys (the arrows), the Space bar, Enter key, Clear key and the Break key. Each of these keys returns a different decimal value if pressed. The up arrow returns an eight, the down 16 and Enter one, hence the variables U,D and E. Quite simply, if the value in A (not *of* A) corresponds with the value of U,D or E, then the next section is executed. As AND statements are tested from the left if the test is not true, for example, in 110 up to D, then the test CP<LL is not carried out; this means that responses are very quick! In fact, if you are a bit heavy fingered the cursor will rush from top to bottom and vice versa, so you may need to put a short delay into line 100 to give you time to get your finger out, or rather off, the key.

Of course, to detect the presence of a particular key depression without destroying the display, you could use the INKEY$ (same as GET$ for other computers) statement but look at Table 1.

```
********************************************************************************
 0                                                                              *
 1              CURSOR CONTROL MENU DEMONSTRATION                               *
 2                                                                              *
 3           FACILITY NUMBER 1                                                  *
 4                                                                              *
 5           FACILITY NUMBER 2                                                  *
 6                                                                              *
 7           FACILITY NUMBER 3                                                  *
 8                                                                              *
 9           FACILITY NUMBER 4                                                  *
10                                                                              *
11           FACILITY NUMBER 5                                                  *
12                                                                              *
13                                                                              *
14      POSITION CURSOR AT REQUIRED FACILITY AND PRESS 'ENTER'                  *
15                                                                              *
********************************************************************************
```

The menu display set up by the program in Listing 1.

very adaptable as a subroutine using different values for U,D,E,CP,LL,TL and MO for different menu displays.

I do not believe it was totally a stroke of luck that Tandy put the cursor control, Enter, Clear, etc, as different bit settings in the same address, but it has made the machine code programs of the Alien Invader type much easier to write as the code only needs to keep checking the bit setting in that single address. Just think how much more difficult and slower it would be if you wanted to use L, R and F to control movement and missile firing — this needs a search routine to check three addresses. Mind you, you could use any three characters which occur in a single address.

THE WAR GAME

To further examine cursor control, horizontally this time, I offer this second program (Anti-Aircraft Interactive Demonstration) which

You can see that if you used a INKEY$ instruction, the in-built machine code routine of the

I/O address (decimal)	Keyboard characters							
14337	@	A	B	C	D	E	F	G
14338	H	I	J	K	L	M	N	O
14340	P	Q	R	S	T	U	V	W
14344	X	Y	Z					
14352	 0	! 1	" 2	# 3	$ 4	% 5	& 6	' 7
14368	(8	) 9	* :	+ ;	< ,	= –	> .	? /
14400	ENTER	CLEAR	BREAK	↑	↓	←	→	SPACE
14464	SHIFT							
INPUT BITS	0	1	2	3	4	5	6	7
DECIMAL VALUE	1	2	4	8	16	32	64	128

Table 1.

interpreter set in train has in strict sequence to examine eight separate addresses for a key depression and then check the value and address against the test criteria. By restricting the routine to just one address you should, in theory, get an eightfold increase in execution time — you do not in BASIC, but you do get a neater routine. This, of course, becomes

demonstrates that interactive games are just about possible in BASIC. It is a simple war game in which a number of enemy aircraft are sent in — one at a time at different speeds and heights. All you have to do is blast them out of the sky with your unlimited supply of ground to air nuclear missiles. Easy!

Taking the program in the order which it runs rather than the order in which it is written, here are the hows, whats and whys.

Line Function
Line 10: Integer variables use less space and time, so use them whenever possible. DEFINT sets all variable names A to Z (whether used or not) to integer variables.
Lines 290 to 330: Just simple instructions; the rather odd spacing between some words is just to ensure that words are not wrapped round one line to the next on display. Line 310 contains some characters (the arrows) which are difficult to print as you are coding, unless you know the trick. It is this: press and hold down **Z**, press and hold down. **2.** Now, back space the cursor to erase the Z and 2 on the screen. Now, press carefully **3**, **4**, **5** or **6** and you will get the four different arrows, and as long as you do not release the Z or 2 you can get as many as you like. Note, however, that this does not work in edit mode.
Line 340: Generates a random number of aircraft, up to 30, as your targets.
Line 360: Displays the number of aircraft as part of the print string; the number fits in where the # # occurs.
Line 370: Flashes the Red Alert warning.
Lines 20 and 30: Assign all the constants used to named variables; this is done to improve the execution time (variables are stored in order as they are assigned and are fetched by the machine much more efficiently than numerical constants). As you already know, L, R and F are assigned the decimal values of the cursor control keys to be used. LL is the left most limit of the missile base at the bottom of the screen, RL is the right most and PP is the initial position of the missile base. MO and MM are the movement variables for the base and missiles respectively. Nn are the other step variables to control the plane's speed and height, and the string representation of the missile base and aircraft.
Lines 40 to 60: The first six values contained in the DATA statements of lines 390 and 400 are assigned to the EX(I) array; this is used to simulate a nuclear explosion whenever an aircraft is hit. The actual screen display is a half screen flashing using almost the same machine code routine as in Listing 1. The two possible values stored in EX(I), 191 or 32, are all pixels on and all pixels off (see reference to pixels in Listing 1 — 191 corresponds to P63). P$ is built up of 11 graphics characters, the last three of which are blanks. TRS-80 graphics characters tie in exactly with the pixel codes given in *Computing Today* with 128 being the same as PO and 191 as P63; 188, for example, is

P60. B$ is the missile base built up of five graphics blocks including blanks at each end. M$ is an up arrow to represent a missile. The blanks are important on moving strings as, without them, printing the string one place left or right would eventually create an elongated display of the outside characters; the blank characters erase this.

Lines 70 and 80: This is the machine code routine and is almost identical to the one in Listing 1 (what was said there still applies).

Line 90: Clears the display and places the missile base towards the right-hand side of the screen on the bottom line; note the ';' to stop the linefeed which would destroy the display.

Lines 100 to 160: Control the speed, height and position of each aircraft. The rather tortuous code is to ensure that each plane emerges from the right-hand side of the screen and leaves by the left-hand side (assuming you do not hit it!). The actual bit of the plane visible is set up by assigning the correct values to S and E and then printing that part of P$ in line 160. F1 is a simple flag used to determine when another plane should be released.

Lines 170 to 200: Used to control the movement of the missile base from side to side and to fire the missiles, one at a time. If no missile is fired, line 200 takes the program back to moving the target plane.

Lines 210 to 230: Control the ascent of the missile towards the target; only if a target is hit or the missile goes off the screen can another one be fired.

Lines 240 to 250: Are used to represent bits falling off the plane and three nuclear explosions — quite effective! The RESET and SET instructions turn off or on randomly selected pixels in the immediate vicinity of the target plane.

IN CONCLUSION

I do not pretend that this game program is the best (it is far from perfect), but it is an interesting demonstration.

A final word of warning, if you ever want to PEEK(14400) with a decimal value of four you *must* disable the break routine first. In BASIC II you can use POKE 16396,165, and this has one advantage in that Shift Break will still work; the re-enable is POKE 16396,201. For Disc BASIC users the normal routine is POKE 23887,23. However, to get Shift Break to work in Disc BASIC, you must POKE 16396,165: POKE 16299,165 but you must re-enable before any disc accesses or you will just end up either with nothing or in DOS. The re-enable command is POKE 16396,195: POKE 16399,195. ■

Program Listing

```
 10  DEFINTA-Z:GOTO 290
 20  A=14400:L=32:R=64:F=8:LL=961:RL=1018:PP=1015:MO=1:
     MM=64
 30  N0=0:N1=1:N2=2:N3=3:N4=11:N5=5:N6=6:N7=18:N8=8:
     N9=10:M1=21
 40  FOR I=1 TO N6:READ EX(I):NEXT
 50  FOR I=1 TO N4:READ X:P$=P$+CHR$(X):NEXT
 60  FOR I=1 TO N5:READ X:B$=B$+CHR$(X):NEXT:M$=CHR$(91)
 70  FOR I=(&HFFF2) TO (&HFFFF):READ X:POKE I,X:NEXT
 80  DEFUSR=(&HFFF2):C$="[SPC]"
 90  CLS:PRINT@PP,B$;
100  FOR P=N1 TO P1
110  SP=RND(N3):H=RND(N8)-1:E=N1:X=MM-E
120  S=N1:F1=N0:IF E=N4 THEN 160 ELSE E=MM-X:GOTO 160
130  X=X-SP:IF X<N0 THEN X=N0:GOTO 120:ELSE 120
140  IF X>N0 THEN 130 ELSE S=S+SP:E=E-SP:
     IF E<N0 THEN E=N1
150  IF S>N9 THEN S=N4:F1=N1
160  AP=X+H*MM:PRINT@AP,MID$(P$,S,E);:IF F1 THEN 260
     ELSE 170
170  IF PEEK(A) AND R AND PP<RL THEN PP=PP+MO:
     PRINT@PP,B$;:GOTO 200
180  IF PEEK(A) AND L AND PP>LL THEN PP=PP-MO:
     PRINT@PP,B$;:GOTO 200
190  IF PEEK(A) AND F AND M=N0 THEN PRINT@PP+N2,M$;:
     M=N1:RP=PP-MM+N2
200  IF M THEN 210 ELSE 140
210  IF RP<N0 THEN M=N0:GOTO 230
220  IF RP-AP<F AND RP-AP>-N1 THEN 240 ELSE PRINT@RP,M$;
230  IF PP=RP+MM-N2 THEN PRINT@PP,B$;:RP=RP-MM:GOTO 140:
     ELSE PRINT@RP+MM,C$;:RP=RP-MM:GOTO 140
240  RP=INT(RP/M1)-N1:FOR I=N1 TO N9:RESET(X*N2+RND(N7),
     RP+RND(N3)):SET(X*N2+RND(N7),RP+RND(N3)):NEXT
250  FOR I=N1 TO N6:POKE(&HFFF6),EX(I):F1=USR0(0):
     FOR J=N1 TO N9:NEXT J:NEXT I:HI=HI+N1:M=N0
260  NEXT P:CLS:PRINT "YOU MANAGED TO DESTROY ";HI;"
     ENEMY PLANES FROM A SQUADRON OF ";P1:PRINT
270  IF HI>(P1/N2) THEN PRINT "DAMAGE AND CASUALTIES
     WERE KEPT TO A MINIMUM":GOTO 410
280  PRINT "EXTREMELY HEAVY DAMAGE AND HIGH CASUALTIES
     INCURRED":PRINT:PRINT "YOUR CITY VIRTUALLY WIPED
     OUT":GOTO 410
290  CLS:PRINT "THE OBJECT OF THIS GAME IS TO ATTEMPT TO
     DESTROY ALL THE ENEMY[2 SPC]AIRCRAFT ENTERING YOUR
     SECTOR."
300  PRINT "EACH ATTACKING SQUADRON OF BOMBERS MAY
     CONTAIN UP TO THIRTY[5 SPC]AIRCRAFT WHICH CAN COME
     INTO THE RANGE OF YOUR ANTI-AIRCRAFT[4 SPC]MISSILES
     AT VARYING HEIGHTS AND SPEEDS.":PRINT
310  PRINT "YOU CAN CONTROL YOUR MISSILE BASE BY
     PRESSING:-[24 SPC]  TO MOVE TO THE RIGHT[42 SPC]
     TO MOVE TO THE LEFT[43 SPC]  TO FIRE A MISSILE":
     PRINT
320  PRINT "YOU CAN ONLY FIRE ONE MISSILE AT A TIME, IF
     YOU MISS RELOADING[2 SPC]IS NOT COMPLETED UNTIL THE
     MISSILE HAS DISAPPEARED FROM THE[5 SPC]SCREEN":
     PRINT
330  PRINT "PRESS 'ENTER' TO CONTINUE";
340  FOR I=1 TO 100:P1=INT((RND(30)+30)/2):NEXT
350  C$="RED ALERT":D$="[9 SPC]":INPUT Z$:CLS
360  PRINT@280,C$;:PRINT@452,USING "MESSAGE FROM AIR
     COMMAND SURVEILLANCE CENTRE, ## AIRCRAFT[8 SPC]
     REPORTED HEADING TOWARDS YOUR SECTOR - ARRIVAL
     IMMINENT";P1
370  FOR I=1 TO 5:PRINT@280,D$;:FOR J=1 TO 100:NEXT J:
     PRINT@280,C$;:FOR J=1 TO 100:NEXT J:NEXT I
380  GOTO 20
390  DATA 191,32,191,32,191,32,176,184,188,188,188,188,
     190,181,32,32,32,32,176,188,176,32
400  DATA 33,0,60,54,0,17,1,60,1,63,2,237,176,201
410  END
```

ENTER THE DRAGON

HARDWARE SPECIFICATION

★ 6809E Microprocessor, a great advance on the original 6502 — still used by PET, Apple, Atom, Atari 400, BBC Micro, VIC 20
★ 32K RAM memory as standard — At least twice as powerful as other computers at the same price, expandable to 64K.
— 26K user available after 4 pages of high resolution graphics.
★ DRAGON 32, unlike most units, gives EXTENDED MICROSOFT COLOUR BASIC as standard
— Microsoft basic has become the industry standard (e.g. IBM, Apple, Commodore, Tandy, Atari)
THIS HAS: -
— Advanced graphics features — set, line, draw, circle, paint, print using.
— Advanced sound feature
— Automatic control of cassette recorder.
— Full editing features — insert, delete, change.
★ DISPLAY:
— 9 Colours.
— 5 Different resolutions from 512 points of text (16 × 32) to 49152 points (256 × 192) at high resolution.
— Home UHF TV set and/or colour monitor
★ KEYBOARD:
— Professional quality keyboard to the standard on data entry terminals.
— Typewriter layout and feel
— Guaranteed for 20 million key depressions
★ SIMPLE PLUG-IN CONNECTION FOR —
— Joystick controllers
— Audio cassette recorder (inc. Stop/Start)
— Printer (Centronics parallel)
— Games cartridges
★ 160 page "BASIC" training manual — FREE

PROGRAMMING STATEMENTS AND COMMAND

MATHEMATICAL AND LOGICAL OPERATORS

Symbol	Operation
↑	Exponentiation
—	Unary minus
★	Multiplication
/	Division
+	Addition
—	Subtraction
>	Greater than
<	Less than
=	Equal to
<>	Not equal to
>=	Greater than or equal to
<=	Less than or equal to
NOT	logical NOT
AND	logical AND
OR	logical OR

BASIC LANGUAGE STATEMENTS

CLEAR	LINE INPUT
CLS	ON . . . GOSUB
DATA	ON . . . GOTO
DEF	POKE
DEFUSR	PRINT
DIM	PRINT TAB
END	PRINT USING
EXEC	PRINT @
FOR TO STEP NEXT	READ
GOSUB	REM
GOTO	RESTORE
IF	RETURN
INPUT	STOP
LET	

SOUND GENERATION STATEMENTS

PLAY SOUND

CASSETTE RECORDER CONTROL STATEMENTS

AUDIO	CLOSE	EOF (−1)	OPEN
CLOAD	CSAVE	INPUT	PRINT
CLOADM	CSAVEM	MOTOR	SKIPF

PRINTER CONTROL STATEMENTS

LLIST OPEN PRINT

SYSTEM COMMANDS

CONT	LIST	RUN
DEL	NEW	TROFF
EDIT	RENUM	TRON

GRAPHICS STATEMENTS

CIRCLE (x,y)	LINE	PCOPY	PUT
COLOUR	PAINT	PMODE	RESET
DRAW	PCLEAR	PRESENT	SCREEN
GET	PCLS	PSET	SET

STRING FUNCTIONS

ASC	INKEY$	LEN	STRING$
CHR$	INSTR	MID$	STR$
HEX$	LEFT$	RIGHT$	VAL

NUMERIC FUNCTIONS

ABS	INT	POINT	SQR
ATN	JOYSTK	POS	TAN
COS	LOG	PPOINT	TIMER
EXP	MEM	RND	USR
FIX	PEEK	SGN	VAPTR

SOFTWARE AVAILABLE NOW!

CARTRIDGES

Ghost Attack — A Pacman type game for one or two players, fifteen skill levels
Berserk — A challenging shooting game for one or two players
Cosmic Invaders — The infamous game on your own set — 15 levels, 1/2 players.
Meteoroids — Pick your way through the treacherous asteroid belt.

CASSETTES

Compendium of Games — A broad range of games to illustrate the abilities of Dragon 32.
Compendium of Applications — A selection of routines to get the most from your Dragon.
Quest — An adventure game — defeat the dreaded morlock.
Madness and the Minotaur — An adult strategy game in real time.
Computavoice — Allow you to instruct Dragon 32 how to speak using phoenetic sounds.
Graphic Animator — Create simple animated colour cartoons on the screen

ALL THIS FOR £195
(VAT INCLUSIVE IN ALL PRICES)

jade

To: **JADE COMPUTERS, MAIL ORDER DIVISON, COOMBEND, RADSTOCK, BATH BA3 3AN.**
TELEPHONE (0761) 32570 FOR MORE DETAILS.

Please send me:
Enter quantity in box
Dragon 32k Colour Computer ☐ £195

CARTRIDGES		CASSETTES	
Ghose Attack ☐	£24.95	Games Compendium ☐	£7.95
Berserk ☐	£19.95	Application Compendium ☐	£7.95
Cosmic Invaders ☐	£19.95	Quest ☐	£7.95
Meteoroids ☐	£19.95	Madness and the Minotaur ☐	£7.95
Joysticks ☐	£19.95	Computavoice ☐	£7.95
Cassette Recorder ☐	£39.95	Graphic Animator ☐	£7.95

I enclose Cheque/PO Number . . . . for £ . . . . made payable to JADE COMPUTERS LIMITED (Plus £4.50 P&P for Great Britain + Northern Ireland).

Name . . . .

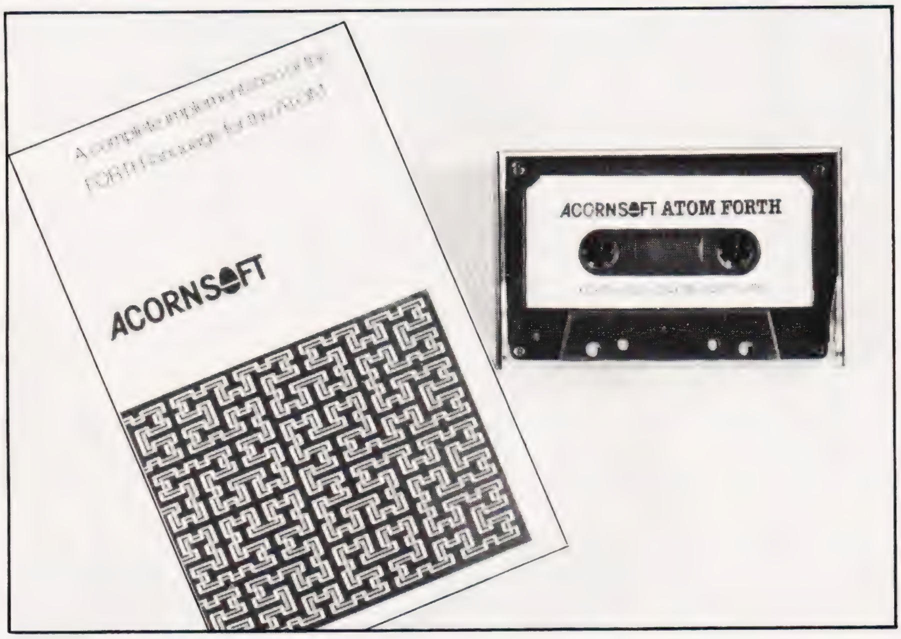
Address . . . .

. . . .

. . . .

ATOM FORTH

Following our popular series on the basics of FORTH as a programming language, we investigate a commercial offering.



ATOM FORTH is a complete implementation of the FORTH language and is based on the standard model described by the FORTH Interest Group. It comes in pre-compiled form on cassette tape and requires an expanded ATOM with a minimum of 12K RAM to operate. However, the Floating Point ROM is not required.

The cassette arrives neatly packaged in the familiar Acornsoft book-like polystyrene box (while looking attractive on a bookshelf, it will probably have to be discarded if you keep your program tapes in a cassette case). The accompanying manual, *FORTH Theory & Practice,* is excellent and is described below.

The FORTH cassette contains:

- Index
- FORTH dictionary and compiler
- Tape Interface and Screen Editor
- Graphics package
- Graphics demonstration

The Index routine is very useful as it allows a visual check on program loading enabling the correct playback volume setting to be obtained. Once the optimum setting has been found (mine worked correctly at the second attempt) FORTH itself can be loaded using the COS command, *** RUN "FORTH"**. Loading takes about five minutes, however, my loading produced a SUM error a few blocks short of finishing. The cassette was rewound a couple of blocks and loading was completed using *** FLOAD "FORTH"**. Executing **LINK # 2800** caused the system to respond with the sign-on message:

```
ATOM FORTH
OK
```

Subsequent re-loadings using different cassette players produced the same SUM error, so presumably the fault was on my tape copy.

THE OPERATING SYSTEM

The ATOM FORTH operating system occupies about 5.5K of memory stored as two blocks. The Lower Dictionary Area is located in block zero RAM from 0240 to 03FE Hex which is normally reserved for BASIC's variables. The Upper and Nucleus Dictionary Areas are located in the entire Lower Text Space from 2800 to 3C00 Hex. This region contains the 'boot-up' parameters used to initiate the system, followed by about 1K of machine code in the form of basic FORTH word definitions. Most of the remaining memory contains a pseudo-code compiled of 'execution addresses' which are used by the FORTH compiler as sequences of addresses of machine code subroutines to be executed. This constitutes much of the pre-compiled Nucleus Dictionary.

FORTH programs, which are more correctly known as applications, are stored in the User Dictionary Area in screen memory between 8C00 and 97C4 Hex. At the very top of the memory map is the User Area (97C4 to 97FF Hex) which is reserved for the 23 operating system variables, five of which are user definable.

UP AND RUNNING

Once running, the words present in the ATOM FORTH dictionary may be viewed by typing **VLIST**. A total of 184 words are included and the listing, which is continuous across the screen, can be halted temporarily by hitting the ESC key. Touching the space bar will recommence the listing.

If you followed D S Peckett's Going FORTH programming series in *CT* earlier this year, you will be familiar with FORTH's use of pre-fix notation. Being a comparative newcomer to FORTH, I found no particular difficulties in handling either it or the stack.

Working through some of the many examples in the manual, I came across two bugs in the FORTH interpreter, concerning the use of the PICK and ROLL words. Both of these words are used to operate on numbers anywhere in the stack; for example, typing **5 PICK** would copy the fifth stack item onto the top of the stack, while **5 ROLL** would rotate the top five stack items. Use of either of these words caused the FORTH interpreter to 'hang-up' with only the Break key being recognised. A quick 'phone call to Acornsoft proved that there was indeed a fault, caused by two incorrect bytes present on early tape copies. A circular arrived in the post the following morning giving correction details:

```
HEX
2B 33C C!
2B 357 C!
```

This sequence was entered and cured the fault. Acornsoft will undertake to replace tapes with this

problem if so wished, however, my replacement tape had the same problem!

The extensibility of FORTH is probably its most distinctive feature. Words may be added to the dictionary by defining their actions with words already present in the dictionary. ATOM FORTH provides six different methods of extending the dictionary — these defining words are:

: (colon)
CONSTANT
VARIABLE
USER
VOCABULARY
CREATE

:, CONSTANT, VARIABLE and VOCABULARY allow words and numerical values to be defined in FORTH terms, while USER provides space for up to five variables, stored in the User Variable Area, thereby allowing system modifications. CREATE is interesting as it can be used to add machine code routines to the dictionary.

As ATOM FORTH does not include an assembler, machine code must be entered in Hex. Using ATOM BASIC's assembler to generate the Hex, which can be done before loading ATOM FORTH, greatly simplifies what could otherwise be quite a tedious chore. A typical use of CREATE is given in Listing 1 which illustrates how the word '2/' could be defined. The compiling words ',' and 'C' compile two or one byte respectively, while SMUDGE is used to allow the new entry to be found in a dictionary search. Another interesting use of CREATE is to call subroutines present in the BASIC ROMs.

ATOM FORTH includes a comprehensive error handling facility with 24 error codes present, six of which are user definable. Error messages are given in the form:

```
cccc  MSG  n
```

where cccc is the erroneous word and n is the error code.

SCREEN EDITOR AND TAPE INTERFACE

Good editing facilities for any language are important and this is particularly so for FORTH. In BASIC, for example, it is possible to examine any particular part of a program simply by listing the relevant line. Similarly, modifications can be made just by re-typing or deleting lines; this is because BASIC programs are stored in source form. In FORTH, however, once a definition has been completed and Return typed, each word is interpreted and then compiled into the dictionary not in source form, but as a series of addresses of machine code routines to be executed when the new definition is encountered. Should any alterations now be required, the word must be deleted from the dictionary and the corrected version re-entered. The ATOM FORTH Screen Editor allows FORTH applications to be stored in 'screens' which preserve the source definition, allowing testing and correction of definitions before finally entering the application into the dictionary or preserving the source form on tape using the Tape Interface.

The Editor and interface are provided in screens #6 to #14 and are loaded (which takes about 5 minutes) by typing **6 LOAD** and playing the tape. As each screen is loaded, it is interpreted and compiled. Each screen is 512 bytes long, rather than the more standar 1024 bytes, and is comprised of eight lines, numbered 0 to 7, each of 64 characters length. The reduced screen size does mean, however, that the entire screen can be displayed on the VDU at one time.

To set up a blank screen for use, the EDITOR word **PROGRAM** is used, and screen numbering may be in the range 0 to 999. The line editor and string editor commands are given in Table 2. To enter a line of text to the screen, the word P is used. For example, to place the definition of TENCOUNT on line three of a screen, the followign sequence would be used:

```
3 P : TENCOUNT 10 0 DO I . LOOP ;
```

The string editor commands allow location, insertion and deletion of character strings. Assuming line 6 of our screen contains 'THIS IS A SILLY DEMONSTRATION' from which we wish to delete the word 'SILLY'. The editing cursor, displayed as #, is first placed at the beginning of the offending word using **F IS A.** The ATOM FORTH response s:

```
THIS IS A #SILLY DEMONSTRATION    6 OK
```

and the word is deleted using:

```
TILL LY
```

which has the result:

```
THIS IS A DEMONSTRATION   6 OK
```

Once a screen is completed, it may be added to the dictionary by typing **ENTER.**

The tape interface provides SAVE, LIST and EMPTY-BUFFERS words which are loaded as part of the EDITOR vocabulary. A screen is put to tape quite simply by typing **SAVE** and replying to the systems prompts. A screen takes about 30 seconds to be SAVEd.

GRAPHICS PACKAGE

The graphics application is provided in screens #18 to #21 and loading, which takes a minute or so, is not dependent on the presence of the Editor. All graphic modes except Mode 4 are available (its screen RAM is used as the User Dictionary Area and is protected from accidental Mode 4 calls).

Like ATOM BASIC, the word CLEAR is used to set up a particular graphics mode with the mode number being specified first in true FORTH style. Points and lines may be drawn using the PLOT, LINE and MOVE words while the plot 'colour' is specified by WHITE, BLACK or INVERT.

To draw a line in white from the origin to a point (15,25) in Mode 2, the following would suffice:

```
2 CLEAR 0 0 MOVE WHITE 15 25 LINE
```

These words all use absolute values for plotting and drawing lines. The words RMOVE, RPLOT and RLINE provide plotting using relative co-ordinate geometry, where the x and y values on the stack are interpreted as being relative to the last point plotted. Also included in the vocabulary is the word REL which converts relative co-ordinates to their absolute values.

The most annoying feature when using the graphics application is the amount of interference present on the display. This is an unavoidable consequence of the use of screen memory for user and application definitions, and can only be avoided by the addition of at least 2K of 'off-board' memory for use as the User Dictionary Area. If the high resolution graphics are not required, it is possible to use their screen memory as extra dictionary space for long or multiple user applications.

The graphics demonstration package uses recursive calls to draw a continuous series of rectangular loops across the Mode 3 screen. It appears on the screen at a stunning rate, greatly illustrating the speed of FORTH.

FORTH THEORY AND PRACTICE

As I've already said, and I make no apologies for repeating myself, the acccmpanying manual is excellent ▶

both as an introduction to FORTH and in providing an insight to its operation. Indeed, I'm sure that even non-ATOM owners with an interest in FORTH would find it fascinating. The 120 page, spiral bound volume is well laid out, easy to follow and contains copious examples throughout, which seem to be remarkably free of errors. The Tape Interface and Screen Editor are fully detailed, however, surely the chapter on graphics could have been extended to more than just three pages? Other chapters cover items such as mathematics, definitions, conditionals and loops, plus a detailed look at the use of the words < BUILDS and DOES> which can be used to define new defining words.

A chapter is also devoted to Further Examples, illustrating the use of recursion, factorals and array sorting. Also included in these examples is an application to patch FORTH into BASIC's COS commands, as well as allowing use of the VIA.

The Glossary details each word present in ATOM FORTH, giving both a description of the word's function plus its stack and system action.

FORTHRIGHT CONCLUSIONS

ATOM FORTH provides, at the relatively low cost of £17.25, a thorough introduction to the FORTH language. Apart from the couple of bugs I encountered (which Acornsoft have now ironed out), it is simple to use and the ease of program development is a delight and leaves me wondering how I manage with BASIC.

Efficient use of the Screen Editor requires some practice, but once mastered presents no difficulties. The graphics words are as easy to use as their BASIC counterparts (if you are prepared to put up with the 'snow' they generate when used).

The manual, which can be purchased seperately for £6 would also not be out of place on any computing bookshelf.

My verdict: if you have an ATOM, get ATOM FORTH.

Finally, my thanks to Acornsoft for answering my questions and allowing me to use a couple of examples from the manual.

FORTH UTILITIES

Lo and behold, no sooner had I handed in this review of ATOM FORTH, than I received a pre-release copy of an ATOM FORTH utilities tape from Acornsoft via our hard working Editor.

The tape, accompanied by a first draft copy of the manual, provides ATOM FORTH with an assembler and discompiler. I was pleased to see the former, though still a little disappointed that it was not included on the FORTH tape itself as it would greatly reduce the tedium of entering all but the shortest of machine code routines

Assembler definitions are enclosed in CODE and END-CODE words in the same way ':' and ';' are used for colon definitions. Nearly all the standard 6502 mnemonics are available and with Hex numbers such as ADC. I say 'nearly all' because there are no branch mnemonics! Branches are performed using loop structures similar to those available in FORTH, eg IF, ...ELSE, ...THEN, (again with terminating commas) and these test specific bits in the status register rather than items on the top of the stack, using the various test words. All addressing modes are available and are typically FORTH in requiring the operand and mode symbol to preceed the mnemonic.

Use of the computation stack is greatly simplified by the inclusion of BOT and SEC. These two words provide access to the low byte of the bottom and second stack items. Using BOT 1+ or SEC 1+ allows the high bytes of the stack items to be accessed. These words effectively make the stack transparent although SEC is somewhat confusing with the assembly language mnemonic SEC,.

Macro Assembly is also catered for, allowing sections of machine code to be used several times, inserting the machine code into the body of the application when it is needed or perhaps until a certain condition is met.

The discompiler does just as its name suggests converting previously compiled words back into an approximate source form. Loading is commenced with **36 LOAD** and takes around four minutes. Acornsoft neglect to inform you that there is not enough RAM for both utilities to reside in the 'normal' system (something I found out after trying to load it!). Resetting the dictionary pointer to a lower address such as 8200 Hex creates more User Dictionary RAM at the expense of High-Res graphics.

Only two words are provided, these are DISCOMPILE and TYPE-OF. Discompiling the previous example of TENCOUNT would produce:

```
LIT #A 0 (DO) I . (LOOP) # -6 ;
```

Address (Hex)	Contents
9800	
	User Variables
97C4	
	Tape Input/Output Buffer
95C0	
	Text Buffer Numeric Conversion Buffer Word Buffer Applications Dictionary
8C00	
	Graphics Mode 3
8600	
	Graphics Mode 2
8400	
	Graphics Mode 1
8200	
	Graphics Mode 0 / Screen VDU
8000	
	Off Board Expansion
3C00	
	Nucleus Dictionary Boot-up Literals
2800	
	Peripherals, etc
0400	
	Graphics Plot Vector
03FE	
	Lower Dictionary Area
0240	
	Operating System Vectors
0200	
	Return Stack
01A4	
	Terminal Input Buffer
0150	
	Free (Econet)
0140	
	COS/DOS Input Buffer
0098	
	FORTH Scratchpad and Pointers
0086	
	Free
0062	
	Graphics Workspace
005A	
	Computation Stack
0000	

ATOM FORTH memory map.

All LITeral values and addresses are given in hexadecimal base.

TYPE-OF returns, not surprisingly, the form of definition along with its execution address. Entering TYPE-OF VLIST returns:

```
COLON DEFINITION  CFA= # 2FA
```

If the word is a constant or variable, its present value is also returned, for example:

```
TYPE-OF  TIB
USER  VARIABLE
VALUE  336
CFA= # 2E0C
```

This utility is extremely addictive, as my wife will testify, so beware!

My draft manual consisted of 20 pages of which 17 were dedicated to detailing the assembler (the manual assumes the reader has a basic knowledge of 6502 assembly language programming). As with *FORTH Theory & Practice*, Dick Harrison has done a good job (other software houses take note) presenting sound, readable documentation.

The FORTH Utilities should be available for purchase now and I stress that this is Acornsoft's prediction, not mine. Cost will be £11.50 (inc VAT) which to my mind is rather expensive. However, it does provide two useful utilities which the FORTHaholic would find difficult to be without. ■

The OFFICE INFORMATION CENTRE

- Word Processing
- Accounting Systems
- Financial Planning

Model II is British Telecom Approved for Datel 200,600 and 2412 Services

Tandy COMPUTER CENTER

Availability - TRS-80™ Computers and accessories available at most of our 300 UK outlets. Some listed below.

Software - Tandy because TRS-80 has the biggest computer library of ready-to-run programs. No experience needed.

Expandability - Buy your ideal sized computer and never replace it. Simply expand it as your needs grow.

Systems From £2699
All Prices Include V.A.T.

Training • Classrooms • On-Site Servicing • Leasing

YOUR NEAREST TANDY COMPUTER CENTRE

BIRMINGHAM *021-643-5368/5362*
Edgbaston *021-455-9175*
BRADFORD *0274-28431/2*
BRISTOL *0272-214721/2*
CROYDON *01-680-1852*
EDINBURGH *031-225-9337/8854*
LIVERPOOL *051-708-0133*
LONDON
Centre Point *01-836-1327*
Buckingham Palace Road *01-828-1423*
Kingsway *01-242-9425*
Bishopsgate *01-626-6688/9*
Seacoal Lane *01-248-5313/4*
MANCHESTER *061-236-2122*
Deansgate *061-832-2242/9314*
ST. HELENS *0744-30774*
SOUTHAMPTON *0703-36492*

TANDY COMPUTER DEALERS

ABERGELE *0745-826020*
ASHFORD *07842-53024*
BRIXHAM *0845-6565*
CAMBRIDGE *0223-65334*
CORK *22954*
DUBLIN *0001-730698*
0001-776270
LONDON *01-388-5271*
RADSTOCK (Nr. Bath) *0761-32570*
WINDSOR *075-35-58072*

Send For Your FREE 1982 TRS-80™ Computer Catalogue

Name ______________________

Address ______________________

____

Postcode ____________ Tel. No. ____________ CT2

Tandy

Tandy Corporation (UK Branch), Tameway Tower, Bridge Street, Walsall, West Midlands. WS1 1LA

Don't let its size fool you.

If anything NewBrain is like the Tardis.

It may look small on the outside, but inside there's an awful lot going on.

It's got the kind of features you'd expect from one of the really big business micros, but at a price of under £200 excluding VAT it won't give you any sleepless nights.

However, let the facts speak for themselves.

<u>You get what you don't pay for.</u>

NewBrain comes with 24K ROM and 32K RAM, most competitors expect you to make do with 16K RAM.

What's more you can expand all the way up to 2 Mbytes, a figure that wouldn't look out of place on a machine costing ten times as much.

We've also given you the choice of 256, 320, 512 and 640 x 250 screen resolution, whereas most only offer a maximum of 256 x 192.

<u>Big enough for your business.</u>

Although NewBrain is as easy as ABC to use (and child's-play to learn to use) this doesn't mean it's a toy.

Far from it.

It comes with ENHANCED ANSI BASIC, which should give you plenty to get your teeth into.

And it'll also take CP/M® so it speaks the same language as all the big business micros, and feels perfectly at home with their software.

NO OTHER MICRO HAS THIS MUCH POWER IN THIS MUCH SIZE FOR THIS MUCH MONEY.

So as a business machine it really comes into its own.

The video allows 40 or 80 characters per line with 25 or 30 lines per page, giving a very professional 2000 or 2400 characters display in all on TV and/or monitor. And the keyboard is full-sized so even if you're all fingers and thumbs you'll still be able to get to grips with NewBrain's excellent editing capabilities.

When it comes to business graphics, things couldn't be easier. With software capabilities that can handle graphs, charts and computer drawings you'll soon be up to things that used to be strictly for the big league.

Answers a growing need.

Although NewBrain, with its optional onboard display, is a truly portable micro, that doesn't stop it becoming the basis of a very powerful system.

The Store Expansion Modules come in packages containing 64K, 128K, 256K or 512K of RAM. So, hook up four of the 512K modules to your machine and you've got 2 Mbytes to play with. Another feature that'll come as a surprise are the two onboard V24 interfaces.

With the aid of the multiple V24 module this allows you to run up to 32 machines at once, all on the same peripherals, saving you a fortune on extras.

The range of peripherals on offer include dot matrix and daisy wheel printers, 9," 12" and 24" monitors plus 5¼" floppy disk drives (100 Kbytes and 1 Mbyte) and 5¼" Winchester drive (6-18 Mbytes).

As we said, this isn't a toy.

It doesn't stop here.

Here are a couple of extras that deserve a special mention.

The first, the Battery Module, means you won't be tied to a 13 amp socket. And, even more importantly, it means you don't have to worry about mains fluctuations wreaking havoc with your programs.

The ROM buffer module gives you a freedom of another sort.

Freedom to expand in a big way. It gives you additional ROM slots, for system software upgrades such as the Z80 Assembler and COMAL, 2 additional V24 ports, analogue ports and parallel ports.

From now on the sky's the limit.

Software that's hard to beat.

A lot of features you'd expect to find on software are actually built into NewBrain so you don't need to worry about screen editing, maths, BASIC and graphics.

However, if you're feeling practical you can always tackle household management, statistics and educational packages. And because NewBrain isn't all work and no play, there's the usual range of mind-bending games to while away spare time.

Waste no more time.

To get hold of NewBrain you need go no further than the coupon at the bottom of the page.

With your order we'll include a hefty instruction manual so you'll know where to start, and a list of peripherals, expansion modules, and software so you'll know where to go next.

Dealer enquiries invited, please contact:–

NewBrain, Grundy Business Systems Ltd., Grundy House, Somerset Road, Teddington TW11 8TD.

Each NewBrain order will include a FREE comprehensive user manual, a catalogue of expansion modules and peripherals, and a detailed list of available software.

Please send me the following:–

Quantity	Item	Price per item (Inc. VAT & p&p)	Total
	NewBrain A	£233.00	
	NewBrain AD with onboard single line display	£267.50	
	Printer	£466.00	
	Monitor 12"	£142.50	
		Total £	

I enclose a cheque/Postal Order for £________ payable to Grundy Business Systems Reader Account. NewBrain, Grundy Business Systems Ltd., Grundy House, Somerset Road, Teddington TW11 8TD.

Please debit my Access Card No:________ my Barclaycard No:________

Signature________

Name________

Address________

____

________Postcode________ CT/1

Registered Number 1522978
VAT Number 358661618
Please allow up to 28 days for delivery.

NEWBRAIN

Henry Budgett

A SHARP REVIEW

Is Sharp's new personal computer really a replacement for the 'K' or just an upgrade?

One of the first and most innovative Japanese companies to market personal computers in the UK must surely be Sharp. Expansion of the company from a calculator base to the world of the micro took place with their MZ-8OK system some 2½ years ago. At the time much comment was made about its overall similarity to the original 2001 PETs; it had a built-in screen and cassette unit and looked rather like a carefully constructed 'copy'.

Apart from their involvement with the hand-held market, where they currently represent the main driving force, Sharp have also produced the MZ-80B and the PC-3201. The former is still horrendously overpriced and the latter proved to be somewhat slower at doing its calculations than should have been expected of a business system. So, what have they been doing to rectify the situation?

The recently launched MZ-80A is what they've been doing and, at first glance, it looks decidely familiar. Fundamentally, it is an upgraded MZ-80K; the screen format has been slightly improved, it now has a decent keyboard and the general systems aspect appears to have been considerably tidied up. But, these changes might be just superficial — what have they done 'under the bonnet'?

HARD OPTIONS

In terms of its internal hardware the 'A' does not exactly break any new ground. Rather, in true Japanese style, it shows the careful and well planned use of the highest level of technology proven to be 100% reliable. The main PCB consists of remarkably few devices, all the main functions being handled by LSI chips like the 8253 and 8255. The UK standard 48K of dynamic RAM is configured from 4116 devices and there's nothing remarkable about them!

Having said that the basic configuration is conventional there are a number of little 'extras' awaiting discovery. The integral display is of the 9" green phosphor type but instead of allowing just the usual 1K of display of memory, Sharp have provided double that. The 1K currently on display is effectively a window on the 2K of memory and can be scrolled up or down. Now, as well as being able to look at those lines of listing that you just scrolled off the top of the screen, you can play all sorts of interesting tricks with double displays — great for games as well as serious programs. A software switch is also provided to lock the screen into the standard 'K' format providing easy program portability.

Being a Z80-based system, the memory map is configured so that the RAM begins at the bottom, 0000 Hex. Now, if you happened to be thinking about offering CP/M as an optional DOS, this is a little inconvenient as CP/M requires the bottom area of RAM for its own obscure purposes — one of the reasons you don't get it on 6502-based systems. Being a forward thinking organisation, Sharp have provided a second software switch which swaps a block of RAM out of the bottom area of memory to make room for the CP/M bootstrap. The user doesn't even know that it has happened as the total amount of RAM is unchanged. The above theory as to why Sharp provided the feature is unconfirmed but it seems to be the only logical one as they are currently offering CP/M with the 'K' and 'B' models.

There is, however, one howling flaw in the internal design. The power supply is mounted on the right of the case, directly under the integral cassette unit and there is no shielding of any kind between the two. In a system as carefully thought out as this, it is an almost unforgiveable error. The cassette system may be a proven and reliable design (it is) but that is still no excuse to subject it to excess heat and stray magnetic fields.

BOLT-ON EXTRAS

If you wished to expand the 'K' you had to take the system bus into an expansion unit, add interface cards to suit and then attach extras such as discs and printer. While technically sound it was a little messy, so someone obviously sat down and had a quiet think. The solution for the 'A' is to have the expansion box, complete with power supply, slide into the rear of the 'A' so that it forms an integral unit. This eliminates one multiway lead and has also allowed the connections to the peripherals to become shielded multiway leads rather than flat ribbon cables as on the 'K'. The overall appearance is much neater and there is no longer any danger of leads being pulled as the connectors chosen — D type plugs and sockets — can be retained by screws.

The range of peripherals is not ▶

vast, three versions of matrix printer and single or dual discs, but sufficient to form a complete system. The DIY enthusiast can add his or her own extras via a universal interface card; doubtless more options will arrive in time.

MONITORING THE SITUATION

The first time user could be easily be forgiven for thinking that although the 'A' has a 4K monitor program, direct machine code access was not possible. Well, it is, but in a very restricted way. The main function of the monitor program is to provide a suite of routines for accessing various parts of the system through machine code calls — all the necessary addresses are given in the manual. Among these routines are facilities for storing data in given memory locations and registers. A full assembler program is available from Sharp as well as the exceptional Zen from a number of independants and this is the more logical route to take if you are interested in machine level programming.

There are just four single letter commands available in the monitor: L for loading a tape into the system; F for booting the discs; J for jumping to a specified address and B for setting or clearing the keyboard bleeper. While this may seem somewhat limited, and it is, Sharp have provided a fully disassembled listing of the monitor program so systems programmers will be able to patch in routines of their own if they so wish.

SOFT TALKING

The 'A' follows the usual Sharp philosophy in that it is a soft machine. To explain more fully, this means that you have a system which has no inbuilt language — you must load in the one you require. Now, while this does take a minute or two, unless you've got discs, it does allow you much more scope for future development; when you tire of BASIC, simply load in Pascal or assembler. It also means that there is no memory overhead, the 48K of RAM could, if you so desired, be filled with machine code — something that you couldn't do if there was a 12K BASIC interpreter sitting in there. Personally, I like this type of approach but some do find it irksome to continually have to reload the BASIC every time you turn the power on.

The standard Sharp BASIC for the 'A' is version SA-5510, a cassette based interpreter. Fundamentally, it is identical to the version supplied with the 'K' except that it uses a different token structure. However, to solve this problem (and the difference in the memory map for the screen) a 'K' to 'A' conversion tape is available which works very well. One little note about this tape is that it converts screen addresses to '%' symbols so it is not a bad idea to use variables instead wherever possible — simply to save time later.

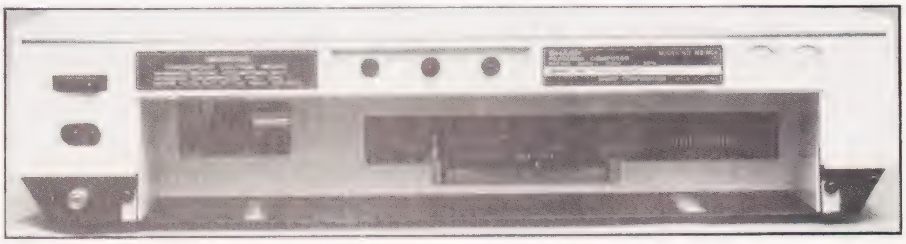
Although the main part of the BASIC is perfectly standard there are a number of nice touches. AUTO line numbering is incorporated (sadly RENUMBER is not) and there is full cassette file support. Rather more I/O functions than usual are supported which reflects the systems approach of Sharp's philosophy. Among the dedicated commands are COPY which dumps the screen to the printer and INP and OUT which receive and send bytes to the specified port. The graphics functions are limited to the pre-defined character set plus the chunky pixel capability of 80 by 50. The music functions of the 'K' have been retained which may, or may not, be a bad thing depending on your point of view (at least the volume control is easily accessed on the 'A'!).

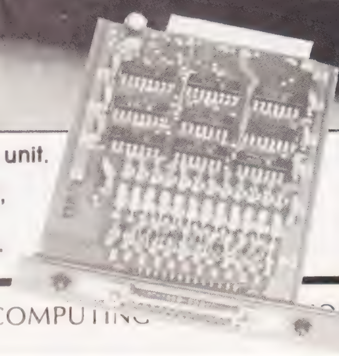
The BASIC is accompanied by a second cassette which contains a number of simple demonstrations, none of which are satisfactorily documented. This is in sharp (sorry about that) contrast to the programs supplied with the other version of BASIC, Disc BASIC. As well as supporting all the Cassette BASIC commands, it also features full error trapping and both sequential and random access disc files. Disc files and programs may be LOCKed against access in order to prevent accidental corruption of data and the almost unique SWAP command is present. This latter command must have been dreamed up by an adventure game enthusiast, so ideally is it suited to the task...it has some serious uses too! For those with restricted ambitions the CHAIN command might suffice.

I mentioned earlier that the disc programs were better documented than their cassette counterparts. This comment is primarily made on the basis of the fact that a complete inventory package showing all the file access possibilities is provided with full instructions and listing. A couple of simple utilities are thrown in as well, including a copy program which allows back-up copies of the master disc to be made — something that wasn't possible on the 'K'. It is interesting to note that the copies of the master cannot then be copied again, an example of watermarking being put to good use, one suspects.

PARLEZ VOUS?

Sadly, one of the areas that the





Top: The recessed slot for the expansion unit.

Bottom: With the expansion unit installed, the rear overhang becomes a little large! (Inset: One of the interface cards).

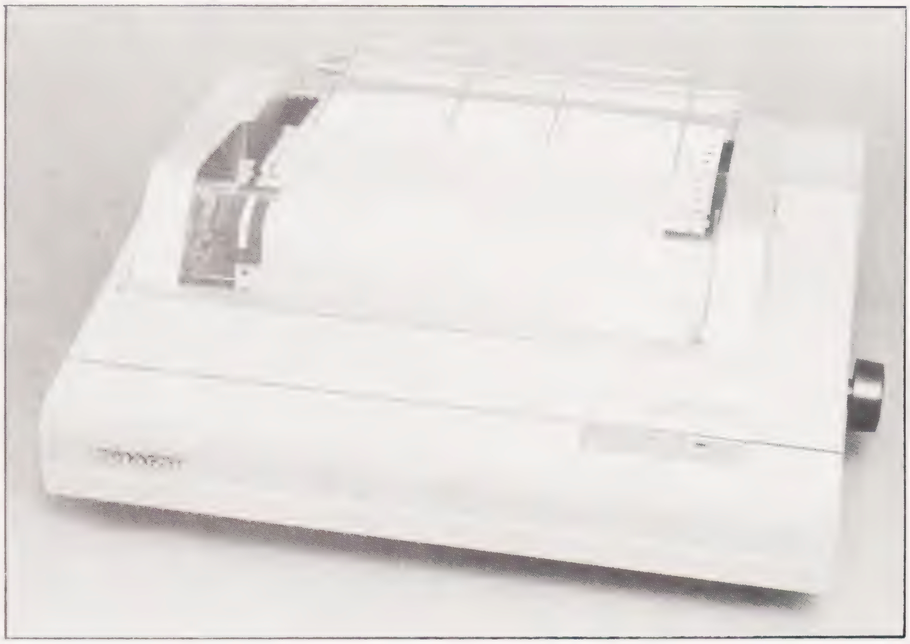
various Japanese companies fall down on every time is the documentation. The old 'K' manuals were rather poor, suffering, as is often the case, from the use of Janglish. While the 'A' manual provides far more useful and interesting information than that offered in the previous volume, it still leaves a lot to be desired.

The first volume is dedicated to the Cassette BASIC and is very familiar. I didn't have the 'K' version to hand but it looks as though the two are more than a little similar. The use of Janglish is common; 'type in a massage' being one of my personal favourites, and there are a number of quite horrific errors in the various example programs. On one page, during a discussion of the various string handling commands, there are no less than three fatal errors — hardly a recommendation to use that translator again.

The Japanese also seem devoted to the use of cartoon characters to illustrate their texts. While I have absolutely no objection to this as a method of getting points home, I do object to the use of such utterly ghastly specimens as found in this manual. If you are going to do something along these lines then please, do it properly or leave it out completely.

These points aside, the manual does contain a wealth of detail: a complete monitor listing, memory maps, block and functional diagrams, useful monitor routines and a whole host more. These are mainly at the back of the book, as is a massive section on the workings of the Z80 CPU — something that could have been safely offered separately.

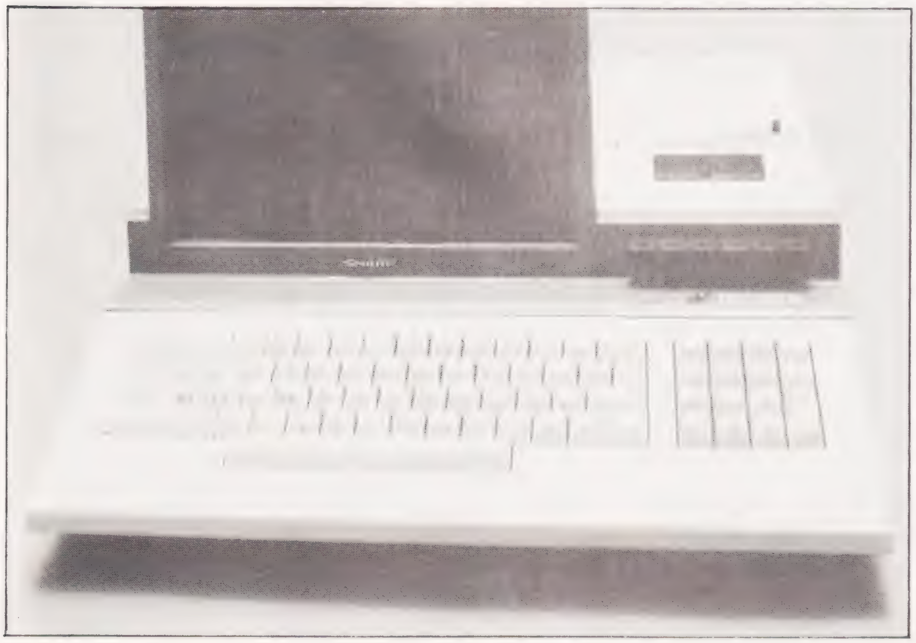
As I was fortunate enough to have the complete system for review there was a second manual on the Disc BASIC and, although it too suffered from the translation problem, it contained most of the information you required.



Overall, the manuals adequately perform their function — there is information for both beginner and serious user, but they often leave one wondering just how they were written — to paraphrase Arthur Dent, "er . . .Ford . . .there's an infinite number of monkeys out here who want to talk to us about this manual they've written for Sharp!".

IN OPERATION

As a system to actually sit down and use, the 'A' is a delight. Like much Japanese equipment it performs faultlessly and predictably; in the many weeks I've had the system running it hasn't once done anything that might cast doubt on its engineering. Having the discs makes life a whole lot simpler, of course, no waiting around for the BASIC to load in from the cassette, but even that operation only takes just over a minute.



The keyboard has an excellent feel and all the keys are where they 'ought' to be. The provision of a 'double zero' key on the numeric pad is a nice touch but why, oh why, did they only provide the '+' and '−' signs and not the other two?

Sharp's monitors have always been excellent and it was nice to see this one holding up the tradition (although the top line of characters had a visible slant to the left on my sample). The colour of the display is green and an anti-glare filter is fitted as standard. The display brightness control is fitted externally on the 'A' (pause for a big cheer as you find that the volume control and a hardware Reset are also located externally too). For those who might be worried about the hard Reset function a J$1250 should perform a BASIC Warm start in Cassette BASIC.

The cassette unit has had its keys repositioned from the version fitted to the 'K', but it looks essentially the same. A tape counter is fitted which makes life that little bit simpler — provided you remember to write the numbers down! As a point of interest, the Cassette BASIC can also be copied from the master supplied with the system but a backup copy cannot be re-copied so do keep a spare.

The two peripherals which came with the 'A', the disc unit and the printer, both performed faultlessly. The former appears to be identical to the unit we have on our 'K', except that it stores more. The▶

printer proved very versatile; it looks rather like an Epson mechanism in a new casing but this wasn't confirmed. One point I tripped up on was the removal of two pairs of transit screws — perhaps a label stuck inside the printer or on its lid would be an idea here. The ribbon is completely encased in a box (except for the free loop which goes under the head) and is easy and clean to change.

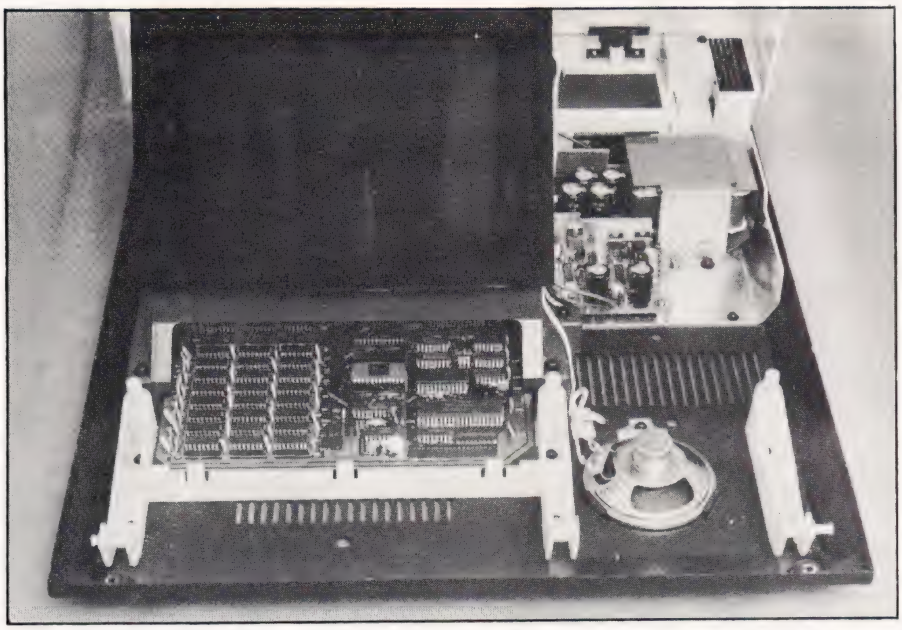
Possibly the only hazard to the use of the system was the depth of the 'A' when its expansion unit was plugged in and the cables attached to the rear. Indeed, the unit needs some 18" ***without*** the expansion unit, so look for a wide desk to site it on (something that you may not think about when you see it in the shop but hits you when you try to find a home for it in the living room!).

THE END PIECE

So, just what is the MZ-80A all about? Well, it's a very nice, usable system which can be expanded to suit most personal and possibly some small business needs. But, and this is a very worrying point, just what have Sharp been doing in their R&D labs for the last two years. Developing the hand-helds is one obvious answer but the 'A' doesn't represent anything like the advance in technology one would expect from this Japanese company.

There are at least three areas of the system which are disapointing, at least when taken in the context of the 'K'. These are: the screen format, the graphics and the overall feeling that I've been here before. However, as a system on its own, it is not too unreasonably priced at the top of the home market and, as is often the case with Sharp's product, it will undoubtedly be offered at a discount price over the next few months.

I must say that I've enjoyed having it around (whether that represents a recommendation I'm not sure) and it is one of only three low-cost home/personal systems with everything built in for less than an arm and a leg (the other two are the ageing MZ-80K and the 4000 series from Commodore). Certainly with the recent price war over the 'K', a number of potential buyers have already purchased the earlier system and at least one I know seems to regret having not waited for the 'A'. Overall then, the 'A' gets the thumbs up but with certain reservations as to just what Sharp think they're doing. After all, you're going to be as sick as the proverbial parrot if they suddenly launch the system we'd all expected the 'A' to be! ■



Internal construction is neat but the unshielded power supply, rear right, is too close to the cassette unit for comfort. Modular construction makes for easy servicing.

FACTSHEET	MZ-80A	
CPU	Z80	
Clock	2MHz	
ROM	6K	
RAM	48K	
Language	BASIC Machine code monitor	
Keyboard	73 key full ASCII Numeric keypad Cursor controls	
Display	25 lines of 40 characters on integral 9" monitor Second page of video RAM, 2K in all 96 block graphics characters Low-Res graphics, 80 by 50 resolution	
Cassette	1200 baud, integral unit	
I/O	System bus connector	
Options	Four-slot expansion unit Twin disc unit, 280K per drive Choice of three matrix printers Universal interface board Disc BASIC	
Costs	MZ-80A	£549.00
	MZ-80AEU Expansion Unit	£115.00
	MZ-80FB dual disc unit (plus disc cable)	£706.60
	MZ-8AFI disc interface	£115.00
	MZ-80P6 printer (plus printer cable)	£496.45
	MZ-8BP5I printer interface	£34.50
	MZ-80I02 universal interface	£51.75
	MZ-8AP5R character generator	£14.95
	Disc BASIC	£35.65
Supplier	Sharp Electronics UK Thorp Road Newton Heath Manchester M10 9BE 061-205 2333	


sinclair
ZX Spectrum
BLUE EDIT
RED CAPS LOCK
MAGENTA TRUE VIDEO
GREEN INV. VIDEO
CYAN
YELLOW
WHITE
GRAPHICS
BLACK DELETE
DEF FN
FN
LINE
OPEN #
CLOSE #
MOVE
ERASE
POINT
CAT
FORMAT
SIN
COS
TAN
INT
RND
STR$
CHR$
CODE
PEEK
TAB
AND
OR
AT
ASN
ACS
ATN
VERIFY
MERGE
IN
OUT
READ
RESTORE
DATA
SGN
ABS
SQR
VAL
LEN
USR
STOP
NOT
STEP
TO
THEN
CIRCLE
VAL $
SCREEN $
ATTR
LN
EXP
LPRINT
LLIST
BIN
IN KEYS
PI
SYMBOL SHIFT
SPACE
BEEP
INK
PAPER
FLASH
BRIGHT
OVER
INVERSE

Sinclair ZX Spectr

16K or 48K RAM... full-size moving-key keyboard... colour and sound... high-resolution graphics...

From only £125!

First, there was the world-beating Sinclair ZX80. The first personal computer for under £100.

Then, the ZX81. With up to 16K RAM available, and the ZX Printer. Giving more power and more flexibility. Together, they've sold over 500,000 so far, to make Sinclair world leaders in personal computing. And the ZX81 remains the ideal low-cost introduction to computing.

Now there's the ZX Spectrum! With up to 48K of RAM. A full-size moving-key keyboard. Vivid colour and sound. High-resolution graphics. And a low price that's unrivalled.

Professional power–personal computer price!

The ZX Spectrum incorporates all the proven features of the ZX81. But its new 16K BASIC ROM dramatically increases your computing power.

You have access to a range of 8 colours for foreground, background and border, together with a sound generator and high-resolution graphics.

You have the facility to support separate data files.

You have a choice of storage capacities (governed by the amount of RAM). 16K of RAM (which you can uprate later to 48K of RAM) or a massive 48K of RAM.

Yet the price of the Spectrum 16K is an amazing £125! Even the popular 48K version costs only £175!

You may decide to begin with the 16K version. If so, you can still return it later for an upgrade. The cost? Around £60.

Ready to use today, easy to expand tomorrow

Your ZX Spectrum comes with a mains adaptor and all the necessary leads to connect to most cassette recorders and TVs (colour or black and white).

Employing Sinclair BASIC (now used in over 500,000 computers worldwide) the ZX Spectrum comes complete with two manuals which together represent a detailed course in BASIC programming. Whether you're a beginner or a competent programmer, you'll find them both of immense help. Depending on your computer experience, you'll quickly be moving into the colourful world of ZX Spectrum professional-level computing.

There's no need to stop there. The ZX Printer–available now–is fully compatible with the ZX Spectrum. And later this year there will be Microdrives for massive amounts of extra on-line storage, plus an RS232 / network interface board.

Key features of the Sinclair ZX Spectrum

- Full colour–8 colours each for foreground, background and border, plus flashing and brightness-intensity control.
- Sound–BEEP command with variable pitch and duration.
- Massive RAM–16K or 48K.
- Full-size moving-key keyboard–all keys at normal typewriter pitch, with repeat facility on each key.
- High-resolution–256 dots horizontally x 192 vertically, each individually addressable for true high-resolution graphics.
- ASCII character set–with upper- and lower-case characters.
- Teletext-compatible–user software can generate 40 characters per line or other settings.
- High speed LOAD & SAVE–16K in 100 seconds via cassette, with VERIFY & MERGE for programs and separate data files.
- Sinclair 16K extended BASIC–incorporating unique 'one-touch' keyword entry, syntax check, and report codes.

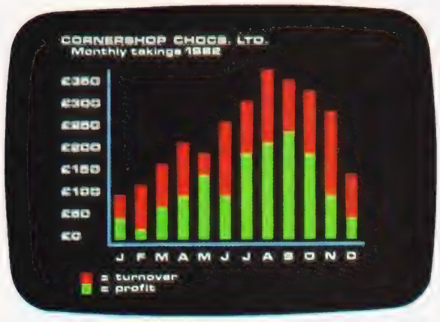
ZX Spectrum software: how good and how soon?

The ZX Spectrum uses an enhanced version of Sinclair BASIC, fast becoming a world standard, and unlikely to be superseded. Unique features, such as one-touch keyword entry and syntax check and report, are increasingly attracting software originators.

Building the software library is already far advanced, and a complete catalogue will be available in the next few months. Subjects will include sophisticated games, education, 'housekeeping', and business management. The more complex packages can, of course, be used to their best advantage with the full 48K RAM version of the ZX Spectrum.

The Sinclair ZX Spectrum can handle sophisticated games programs with high-resolution colour graphics and sound.

This major advance in computer technology maintains Britain's world-beating position in the field of personal computers.



A range of business software will soon be available, covering both specific applications (eg stock-control and payroll) and general business management systems (eg matrix models).

This second generation of Sinclair personal computers demonstrates continuing commitment. Advanced technology made the ZX80/81 family a price breakthrough: advanced technology makes the ZX Spectrum a breakthrough in price and performance.

Elegant, effective, unique—the ZX Spectrum design.

'Less than half the price of its nearest competitor–and more powerful.'

'These two pictures show how it's done. On the right is the PCB from the BBC Model A Microcomputer. On the left is the PCB from the ZX Spectrum.

'It's obvious at a glance that the design of the Spectrum is more elegant. What may not be so obvious is that it also provides more power.

'The ZX Spectrum has more usable RAM, and higher maximum RAM.

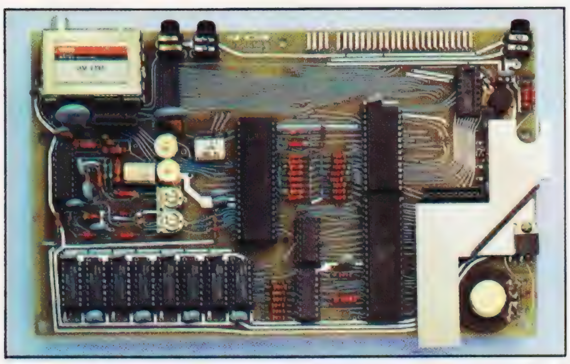
'It offers twice as many colours on the screen at any one time, plus a colour brightness control. It also offers user-definable graphics.

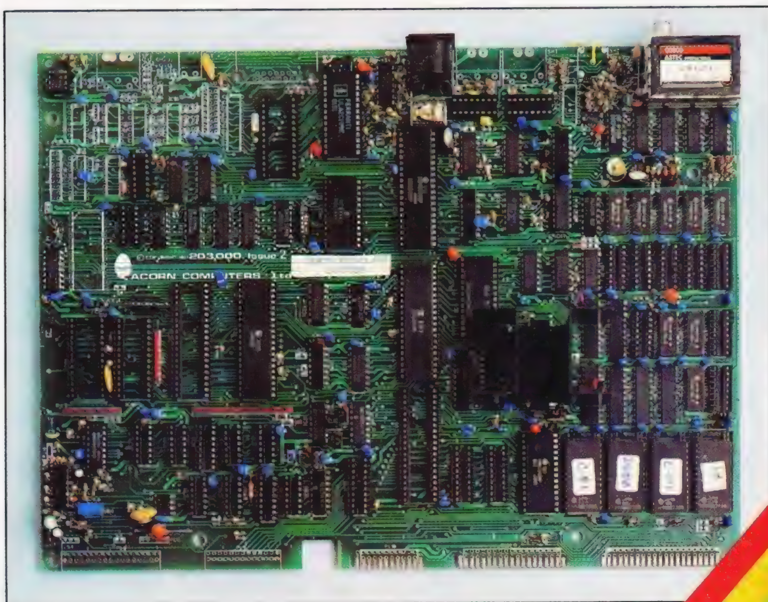
'It has data transfer rate 25% faster, supported by a VERIFY facility.

'And it employs a dialect of BASIC (Sinclair BASIC) already in use in over 500,000 computers worldwide.

'We believe the BBC make the world's best TV programmes–and that Sinclair make the world's best computers!'

–Clive Sinclair.





Above left: internal layout of Sinclair ZX Spectrum.

Right: Internal layout of BBC Micro Model A.

The illustrations are to the same scale, and demonstrate the rate of advance in microcomputer design. The ZX Spectrum uses just 14 chips to provide more power and more user-available RAM.

sinclair ZX Spectrum

F Hooper

TBUG FROM BASIC

Gain access to your Tandy's monitor while programming in BASIC.

If you are the owner of a TRS-80 system and have been brave enough to dabble in machine code, you will probably be well aware of the usefulness of Tandy's Z80 monitor program TBUG. However, for those among you who have not yet heard of TBUG, it is a powerful machine code program designed specifically to create, modify, debug, load and run machine code while allowing complete access to all the Z80 registers (all the registers are preserved intact in the register save area when the program is called).

The main problem with TBUG is one of location. It resides in low memory, occupying locations 4380 Hex through to 497F Hex (the TBUG program actually ends at location 4824 Hex, the area 4825 Hex to 497F Hex is occupied by a massive TBUG maintained stack). This location is too low to be protected by the MEM SIZE? function and any but the most trivial of BASIC programs will overwrite it. Relocating TBUG in high memory, even if easily done, would still not be of any great help as to access TBUG it is necessary first to halt execution and then to use the system call command, which apart from being time consuming also destroys the contents of the Z80 registers at the point at which we would most like them preserved, ie in the middle of a program run.

The solutions to these two problems are, first, to alter all the necessary start of BASIC program text and BASIC variable pointers so that any BASIC programs can be stored in memory locations above the end of TBUG. Second, one of the redundant Disc BASIC terms, in this case GET, is used to call TBUG. GET can be used both in the command mode or as a program statement as will be shown later. I make no apologies for using the Disc BASIC term as TBUG cannot actually be used when Disc BASIC is in operation as Disc BASIC completely overwrites the TBUG location.

When in the command mode the program allows TBUG to be accessed simply by typing 'GET' and pressing 'ENTER'. TBUG can then be used to debug machine code programs and BASIC programs and BASIC programs with embedded machine code routines. The use of this program also creates a new, small but safe area between 42E9 Hex, the old start of BASIC program text, and 4319 Hex, the byte prior to the start of the altered TBUG program. It is perfectly safe to store a machine code routine in these 48 bytes of memory without any fear of overwriting them.

While it is possible to use the modified TBUG program to write machine code directly into dummy strings or dummy program lines, it is not advisable to attempt this. If your machine code program contains a zero byte it will be taken by the BASIC interpreter as an end of BASIC line marker and the following two bytes will then be altered to point to the next line of BASIC, which in this case would be the rest of your machine code. This would occur at every zero byte in the machine code routine with disastrous results. However, if the program contained no zeroes, the use of TBUG can make savings in both programming time and length.

Further savings can be made when writing BASIC programs which use strings for the display of graphics for, rather than writing all the long and laborious CHR$ functions to build up the string, a dummy string can be loaded directly with the required graphic blocks. The string literal is first located within the BASIC program text storage area and then the Hex representation of the required graphic block is entered using TBUG's memory change function. These graphics can then be displayed using any of the normal print functions.

A LITTLE KNOWLEDGE

Before one starts to use the new GET function, it is important to understand a little about the manner in which the TRS-80 stores BASIC programs and variables. BASIC program text is stored at

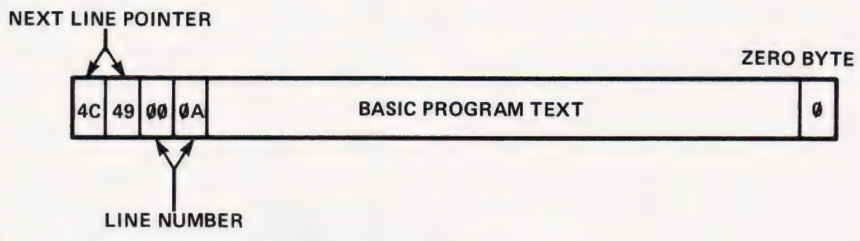


Fig. 1. How the Tandy stores its Basic program lines.

4981 Hex (18817 decimal) onwards and each line of BASIC is stored in the set format illustrated in Fig. 1.

The first two bytes point to the next line of BASIC and the second pair give the line number of the following line of BASIC text. All BASIC commands are stored as tokens and are listed in the user manual; other than these the entire text is stored as a direct ASCII equivalent of the typed line. Finally, the last byte in each line is a zero indicating the end of the line.

The TRS-80 stores variables in a table located at the end of the BASIC program text; the exact location of this table is given by the address stored in the variable table pointer 40F9 Hex/40FA Hex. The first byte in the table is a type declaration byte:

- 02 indicates an integer variable.
- 03 indicates a string variable.
- 04 indicates a single precision variable.
- 08 indicates a double precision variable.

This byte also tells us the number of bytes used to store the actual ▶

value of the variable. The two bytes following the type declaration byte contain the ASCII equivalent of the variable name (this is stored in reverse order — if the variable name was AB then these two would be 42 Hex and 41 Hex respectively). The next byte is the one pointed to by the VARPTR function and is the start of the variable value. The rules governing the storage of variables are adequately covered in the TRS-80 User Manual and so will not be discussed here. Suffice to say that it is far easier to study the variable table using GET than with the VARPTR function.

ONE JUMP AHEAD

The program itself is quite straightforward in operation but a few points do need clarification. Location 41C4 Hex is a BASIC intercept address. The BASIC interpreter calls this location prior to executing a line of BASIC (normally 41C4 Hex contains a return instruction and the following two bytes are zeroed). By placing a jump command at this location we can effectively append any desired routine to the interpreter. The insertion of the jump is achieved by lines 100-150.

Lines 160-270 change the 'Start of BASIC', the variable, the array and the free space pointers, so that the TBUG routine will not be overwritten. Location 417F Hex is the point at which Disc BASIC intercepts the GET command (when Disc BASIC is not being used, there is a command to jump to an error routine here). By changing the jump vector we can intercept the GET command and use it to jump to TBUG—lines 280-300 change the jump vector and lines 310-340 instigate the jump. The jump is not straight to TBUG for two reasons:

1) to standardise the method by which TBUG is exited and,

2) to save the contents of the SP register. This is for continuity of use of the stack and also to allow investigation of the stack used by the BASIC interpreter.

The return address used in line 320 returns the user to the command mode on exiting TBUG. Lines 350-480 are used to intercept the GET instruction when placed in a program line.

As stated before, the BASIC interpreter is forced to jump to this point prior to executing a line of BASIC; the HL register pair will point to either the start of the line or the delimiter in a multistatement line (this value of HL is saved in case it is needed again). RST 10 Hex points to a routine in ROM which will advance the HL register pair to point to the next non-zero character in the line, ignoring any spaces. The BASIC token for the GET function is the Hex number A4 and if the HL register pair do not point to the GET character then control is passed back to the BASIC interpreter with the HL registers restored (lines 370-390 and 490-500). If, however, the GET instruction is found we move on to line 400—here the next character in the line is located, again using the RST 10 Hex call. In the program text this should be the delimiter (:) to avoid any syntax errors which may occur using the GET function in this manner.

The new next line position (HL) is stored so that execution of the BASIC program may continue after exiting from TBUG. The return address used in line 420 is not the address from which the routine is called — a return to which would cause the display of the L3 ERROR message — but rather a convenient place to re-enter the BASIC interpreter. At this point the interpreter expects the HL pair to point to the next line of BASIC to be interpreted, hence the call to RST 10 Hex in line 400.

Lines 440-460 simply tidy up the stack, line 440 removing the unwanted 'Start of BASIC' line address pushed to the stack in line 350. Lines 450 and 460 remove the return address of the initial call to the intercept address 41C4 Hex. Line 470 saves the address of the stack used by the BASIC interpreter so that it may be examined if required and finally, line 480 executes the jump to TBUG.

The last section of the program, lines 510-550, restarts the running of the BASIC program at the instruction following the GET command. Line 510 restores the BASIC interpreter stack, lines 520-530 deposit the return address on the stack, line 540 sets the HL register pair to point to the next line of BASIC to be interpreted and finally, line 550 initiates the restart of the BASIC program.

CREATING A NEW TAPE

The easiest method of creating the adapted TBUG tape is to load and run TBUG as normal and then,

Program Listing

```
100   431A 21 C4 41     START  LD HL,41C4    BASIC intercept
110   431D 36 C3               LD (HL),ØC3   address
120   43IF 23                  INC HL
130   432Ø 36 51               LD (HL),51    Jump vector
140   4322 23                  INC HL
150   4323 36 43               LD (HL),43    Jump vector
160   4325 21 8Ø 49            LD HL,498Ø    HL points to byte
170   4328 AF                  XOR A         before BASIC text
180   4329 77                  LD (HL),A     Zero byte before
190   432A 23                  INC HL        BASIC
200   432B 77                  LD (HL),A     Zero this one too
210   432C 22 A4 4Ø            LD (4ØA4),HL Reset start pointer
220   432F 23                  INC HL
230   433Ø 77                  LD (HL),A     Zero this one too
240   4331 23                  INC HL
250   4332 22 F9 4Ø            LD (4ØF9),HL Reset end pointer
260   4335 22 FB 4Ø            LD (4ØFB),HL Set array pointer
270   4338 22 FD 4Ø            LD (4ØFD),HL Set free memory
280   433B 21 44 43            LD HL,GET     GET = entry point
290   433E 22 8Ø 41            LD (418Ø),HL Change jump vector
300   4341 C3 CC Ø6            JP Ø6CC       Jump to command
310   4344 ED 73 7A 43  GET    LD (SAVESP),SP
320   4348 21 CC Ø6            LD HL,Ø6CC    HL=return address
330   434B 22 7C 43            LD (RETADD),HL
340   434E C3 AØ 43            JP 43AØ       Go to TBUG
350   4351 E5           BASSCH PUSH HL       Save line position
360   4352 D7                  RST 1Ø        Get next character
370   4353 7E                  LD A,(HL)
380   4354 FE A4               CP ØA4        Is it GET token?
390   4356 2Ø 14               JR NZ,GBCK
400   4358 D7                  RST 1Ø
410   4359 22 7E 43            LD (NXLNPO),HL
420   435C 21 5A 1D            LD HL,1D5A    HL=BASIC entrypoint
430   435F 22 7C 43            LD (RETADD),HL
440   4362 E1                  POP HL        Tidy up the
450   4363 33                  INC SP        stack
460   4364 33                  INC SP
470   4365 ED 73 7A 43         LD (SAVESP),SP
480   4369 C3 AØ 43            JP 43AØ       Go to TBUG
490   436C E1           GBCK   POP HL        Restore line posn
500   436D C9                  RET           Go back to BASIC
510   436E ED 7B 7A 43  CONT   LD SP,(SAVESP)
520   4372 2A 7C 43            LD HL,(RETADD)
530   4375 E5                  PUSH HL       Return address
540   4376 2A 7E 43            LD HL,(NXLNPO)
550   4379 C9                  RET           Return to BASIC
560   437A ØØ ØØ        SAVESP DEFW Ø        Stack save area
570   437C CC Ø6        RETADD DEFW Ø6CC     Return address
580   437E ØØ ØØ        NXLNPO DEFW Ø        Line pointer
```

using the memory change function M, write the program machine code starting at memory location 431A Hex. After checking that the program has been loaded correctly, use the punch command:

```
P 431A 4824 431A
(NAME OF YOUR CHOICE)
```

to save a copy to tape. Having done this you can load and run the system tape you have just created as normal or you can initialise the GET function straight away by using the command:

```
J 431A
```

This will immediately return you to the command mode with the GET function ready to use.

An alternative method is to use the Tandy Editor/Assembler to create a system tape, load TBUG and use its load function L to load the system tape, then using the punch instruction, create the altered TBUG tape.

GETTING TO KNOW YOU

As stated earlier, when in the command mode, TBUG can be simply called by typing 'GET' and pressing 'ENTER'. The GET function may also be used as a program statement, for debugging purposes, in one of two ways. First, if you require the BASIC program to continue execution after using TBUG then the format for the GET statement is :GET: . Any of the following are permissable:

```
10  :GET
20  X=15:GET:
30  V=30:GET:FOR Y=.....
```

If, however, you wish to be returned to the command mode after executing the GET call then the following formats should be used:

```
10  GET
20  GET:X=4:Y=.....
```

When you wish to leave TBUG it should always be done using a jump to location 436E Hex. However, should you not wish to continue with the execution of a BASIC program, a jump to location 06CC Hex will return you to the command mode. Under no circumstances should TBUG be exited using a jump to 0000 as the GET function will have to be reinstated. In fact, the GET function will require initialising whenever the MEM SIZE? or MEMORY SIZE? call is displayed although as long as the error which forced the MEM SIZE? call has not overwritten the machine code routine, the GET function can be reinstated by:

```
SYSTEM     <ENTER>
? /17178   <ENTER>
```

and if all is OK the machine will respond with the ready signal.

It is worth noting that any desired routine can be used in place of TBUG, the only criterion being that the routine should be entered at location 43A0 Hex and it should not exceed memory location 497F Hex — a few minor alterations to lines 160 and 480 will, however, overcome even these limitations. ■

HEX	Decimal	Use
40A0	16544	Pointer to end of string space.
40D6	16598	Pointer to current position in string space.
40A2	16546	Holds number of current line being processed.
40A4	16548	Points to start of BASIC text.
40E6	16614	Points to item in BASIC text currently being processed.
40F9	16633	Points to end of BASIC program text and beginning of variable table.
40FB	16635	Array pointer.
40FD	16637	Start of free space pointer.

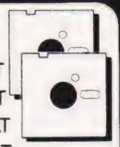
Micro-Spares VALUE

SUPPLIERS TO TRADE
LOCAL GOVERNMENT
EDUCATION
INDIVIDUALS
INDUSTRY

COMPUTERS
PERIPHERALS
COMPONENTS
& NATIONWIDE
MAINTENANCE

19 ROSEBURN TERRACE, EDINBURGH EH12 5NG
031-337 5611

DISKS & TAPES



- 5¼" SSSD BASF **£17.95** + VAT
- 5¼" SSDD BASF **£21.45** + VAT
- 5¼" DSDD BASF **£25.95** + VAT
- 5¼" Cleaning Set **£16.50** + VAT
- 5¼" Library Case **£1.90** + VAT

Cassettes (C20) 65p All storage media is top quality— No High St. rubbish.

Add £1.50 p. & p. per box.
28 Disk protection folder £10.49 + VAT

NEW! Only £155 + VAT.

IN STOCK now
The new colour board from Lucas
nascom
Micro-Spares VALUE

nascom PRODUCTS

"Interest Free Credit Available"

KITS	
Nascom 1, with NAS-SYS 1 less P10	£112.50
Nascom 2, no user RAM	£202.50
BOARD LEVEL	
Nascom 1, with NAS—SYS 1 less P10	£126.00
Nascom 2, no user RAM	£238.50
CASED SYSTEMS	
Nascom 3, no user RAM	£338.40
8K user RAM	£36.00
16K user RAM	£90.50
32K user RAM	£103.50
48K user RAM	£117.00
POWER SUPPLY	
Kit form	£29.95
MEMORY CARDS	
RAM B memory card with 16K RAM—kit	£72.00
RAM B memory card with 16K RAM board	£90.00
Additional 16K RAM	£13.50
Additional 32K RAM	£27.00
I/O BOARDS	
I/O boards for 3 x P10 1 x CTC, 1 x UART (kit) ex P10	£40.50
P10 for above I/O	£10.80
CTC for above I/O	£12.60
UART for above I/O	£14.40

DISC SYSTEMS	
Nascom single disc drive (350KB) incl. FDC card	£423.00
Nascom dual disc drive (350KB each) incl. FDC card	£616.50
NAS DOS disc op system	£40.50
SOFTWARE	
NAS-SYS 1 ROM	£10.80
NAS-SYS 3 EPROM	£18.00
ZEAP 2.1 for NAS SYS in 4 x EPROM	£26.30
ZEAP 2.1 for NAS SYS on tape	£22.50
BK microsoft basic in ROM	£18.00

PAYMENT AND DELIVERY

Payment is by Cheque, Postal Order, ACCESS, VISA etc. PLEASE add postage and VAT. All in stock items sent same day. All non Kit items have a 1 year guarantee.

ALL PRICES APPLY TO END OCTOBER 1982

COMPONENTS

MEMS	
4116 (200nS)	73p
2708	£1.50
2716 (5V RAIL)	£2.11
2114L	95p

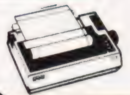
CONNECTORS	
25W plug	£1.50
25W socket	£1.89
Covers	£1.20

TOP VALUE PRINTERS TOP VALUE

Anadex DP8000 B & O Matrix	£300 + VAT
Tec 45 & 55 Cps Daisy Wheel	£995 + VAT
Silver Reed Typewriter/Printer RS232	£500 + VAT
RICOH RP1600	£1149 + VAT
Triumph-Adler Stylist	£595

EPSON PRINTERS PRINTERS

from £307



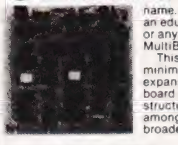
MX80FT—I	£307 + VAT
MX80FT—II	£315 + VAT
MX80FT—III	£327 + VAT
MX100 Type	£439 + VAT
MX82FT	£330 + VAT

TERMINALS/MONITORS

BMC 12v Green Screen Monitor	£119 + VAT
Televideo 910 Terminal	£425
Televideo 925 Terminal	525
Televideo 950 Terminal	£615

1 YEAR GUARANTEE ON ALL NON KIT ITEMS

GEMINI



The Gemini MultiBoard concept is the logical route to virtually any microcomputer system you care to name. Whether you require a business system, an educational system, a process control system or any other system, there is a combination of MultiBoards to fulfil that function.

This concept ensures maximum flexibility and minimal obsolescence. Maintenance and expansion is greatly enhanced by the modular board design. MultiBoard is based on the 80-BUS structure, which is finding increasing acceptance among other British manufacturers; thus broadening the product base.

HARDWARE (BUILT & TESTED)

GM802	64K RAM card	£140
GM803	EPROM ROM card	£65
GM807	3A PSU	£40
GM808K*	EPROM programmer	£29.50
GM809	FDC card	£125
GM810K	5A PSU 8 slot motherboard	£69.50
GM811	Z80 CPU card	£125
GM812	Z80 IVC card	£140
GM813	Z80 CPU/64K RAM card	£225
EV814	IEEE 488 card	£140
GM815-1	Single drive disk unit with PSU (350K)	£325
GM815-2	Double drive disk unit with PSU (700K)	£550
GM816	Multi I/O board	£140
AM819	Speech board	£85
AM820	Light Pen	£35
GM821	*SC II keyboard	£57.50

(*Kit)

SOFTWARE

GM512	CP/M 2/2 for Multiboard	£90
GM517	Gem-Zap edit/asm tape	£45
GM518	Gem-Zap edit/asm disk	£45
GM519	Gem Pen editor text formatter tape	£45
GM520	Gem Pen editor text formatter EPROM	£45
GM521	Gem Pen editor text formatter disk	£45
GM524	Gem Dis disassembler/ debugger tape	£30
GM525	Gem Dis disassembler/ debugger disk	£30
GM526	Comal-80 tape	£100
GM527	Comal-80 disk	£100
GM528	APL disk	£200

THE GALAXY

COMPUTER FOR BUSINESS ETC

Hardware
- Twin Z80A CP/M System
- 64K Dynamic RAM
- 800K Disk Storage (Formatted)
- 80 x 255 Screen Format
- Inverse Video
- Prog. Character Generator
- 160 x 75 Pixel Graphics
- Centronics Parallel I/O
- RS232 I/O
- Light pen interface
- 59-Key ASCII Keyboard

Software
- Full 64K CP/M 2.2 with screen edit facility
- Comal-80 structured BASIC
- GEM-ZAP Assembler/ Editor
- GEM-PEN Text editor
- GEM-DE BUG debugging software

Galaxy System	£1450 + VAT
Green Screen Monitor	£117 + VAT

ANOTHER 'PLUS' FACTOR FOR 'APPLESOFT' USERS

THERE IS NO MORE NEED TO COMPROMISE WHEN WRITING BASIC

Write your program with EXTENSIVE REMS, SINGLE LINE STATEMENTS, FULL LENGTH VARIABLES

For easy readaibility, easy debugging and easy maintenance. Then convert into the shortest, fastes form with **TOOLBOX**

TOOLBOX is a suite of four programs, consisting of the following:

REM REMOVER: removes rem statements from your program

COMPRESSOR: rearranges into multi-line statements of the maximum possible length, taking into account if/then statements, GOSUBS, GOTOS, Returns, etc.

REDUCER: reduces variables to a maximum length of two digits, and also renumbers in increments of one.

RENUMBER: renumbers in increments of ten.

3.3DOS **£18.00** Tape **£16.40**

Also shape generator

Generates shape tables of up to 300 moves, with no other limitation on size.

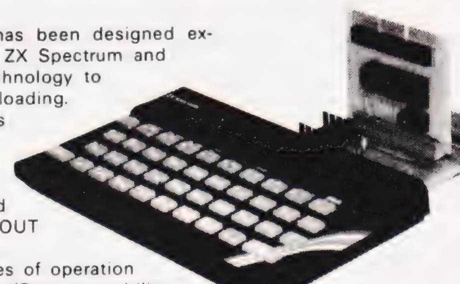
3.3DOS **£15.00** Tapes **£13.40**

From KEYSOFT
24 Wakehams Hill, Pinner, Middx. HA5 3BQ

State memory size when ordering

ZX SPECTRUM HARDWARE

24 line programmable Input/Output Port



This new port has been designed exclusively for the ZX Spectrum and utilises MOS technology to minimise bus loading.

The main features are:

- 3 x 8 bit ports
- Port mapped using IN and OUT commands
- 3 basic modes of operation
- Direct Bit Set/Reset capability
- Outputs capable of sourcing 1mA at 1.5 volts
- Access via 3 x 16 pin DIL sockets **and** 28 way edge connector

The port is available fully built together with a set of detailed instructions and suggested control applications. The port can either be used with our new Motherboard thus allowing a further card to be used, or with a stackable connector. Either way, it means that Micro-Drives, Printers etc. will run happily with the port.

ZX Spectrum PPI Port	£16.50
ZX 2 Slot Motherboard	£16.95
ZX Stackable Connector	£5.50

The prices are inclusive of VAT but postage must be added at 70 pence for a single item (100 pence for 2 or more items).

Please note that we shall continue to support the ZX81 and that we also offer a 16 bit port for the 81 based on a Z80A PIO at £15.50 inc.

KEMPSTON ELECTRONICS
60 Adamson Court, Hillgrounds Road, Kempston, Beds MK42 8QZ

Paul Kriwaczek

MINI-CALC

A spreadsheet program in less than 2K! Not impossible if you use your BASIC wisely.

It's very impressive, but what do you use it for? If you own a personal microcomputer, you're probably as bored with that question as I am. And you probably find it as difficult to give a satisfactory answer.

The truth is that for most of us, the interest of a home micro is in the programming, not so much in using the program once it is written. Of course, there are those who are reputed to keep their home accounts on the computer and to maintain a stock list of what they have in the freezer, but I've never actually ***met*** one. I once asked the chairman of one of Britain's leading computer software houses whether he had a micro at home. He said "If my private life was so damn complicated that I needed a computer to organise it, I wouldn't get a computer, I'd change my lifestyle."!

In fact, my guess would be that most amateur programmers spend their time writing systems software, which, once written, is actually used infrequently, if at all. The situation is not unlike that of the radio ham, or even CB freak, who labours over the technicalities of communication, but once in contact can rarely find much worth communicating.

In the business world too, the personal micro is still a relative rarity. Many company accounts are computerised, of course, and more and more businesses, some of them very small indeed, use a micro to control stock, look after orders and invoices and perform general data management tasks.

But these are company jobs, not personal ones, and few managers have so far seen the micro as being of any direct use to them in the course of their day-to-day work. It is a surprising fact, in the light of the power and flexibility of today's desktop computers that they are yet to be found on so few desk tops.

However, it is becoming clear that at least one kind of software package is beginning to find its niche as a piece of everyday working equipment and that is the word processor package plus printer which is set to replace the electric typewriter as a standard tool for anyone who does any writing — and even some who don't. Given a little more time, and if male executives can give up their deep aversion to touching a typewriter-style keyboard (that symbol of menial femininity) there will hardly be an office desk without its Wordstar, Scripsit or Electric Pencil. The advantages they offer are obvious.

SPREADING THE LOAD

If the word processor is here today, the next application of this technology to become commonplace is already waiting in the wings. It is the kind of package known as a 'spreadsheet' and it is interesting to note that it does for paper, pencil and pocket calculator just what the word processor does for the typewriter.

The first, and still very popular, 'spreadsheet' is a package called VisiCalc. Indeed, its success made it the first piece of software to sell hardware — people bought Apples just so that they could run it. Today there are other packages on the market, some very powerfully enhanced, but they are all based on the same apparently simple idea.

The computer's display becomes a window onto part of a large matrix of memory locations, like a piece of engineers' graph paper, on which is written the content of each square. Every position in the grid can hold either a numeric value, a string or (and this gives the system its power) an arithmetic or logical expression relating it with other points on the matrix.

Part of a 'spreadsheet' display might look like this:

	1	2	3	4
A	MONTH		SALES	
B	JAN		204	
C	FEB		225	
D	MARCH		239	
E	APRIL		217	
F	------		-----	
G	TOTAL		865	
H	------		-----	
I	AVERAGE		216.25	
J	------		-----	
K				

Here, the rows are labelled with letters of the alphabet and the columns with numbers. Each position in the matrix is known by the combination of row and column which intersect at that point. Thus, the top left-hand cell is called A1, to its right is A2 and below it is B1. In the above example, rows A, F, H and J, as well as the whole of column 1, contain strings serving as labels or separators for the rows of the next-but-one column. From row B to row E, column 3 contains values entered by the user. Position G3 would contain the expression B3+C3+D3+E3, to produce the sum, while I3 would be set to G3/4, to give the average. If any of the values in B3, C3, D3 or E3 are changed, this is immediately reflected by new values for G3 and I3.

It is possible to set up very complex calculations and, while changing any of the initial

parameters, instantly observe the effect on the results. Entering a large number of expressions individually can be avoided by using a facility known as replication, which generates a whole row or column of expressions at a time. The relevant rows of column 4 in the above example could all be set to produce the square of the value in column 3 by entering a single command.

This kind of 'spreadsheet' is extremely useful for any kind of calculation which is more than a quick back-of-an-envelope job, whether financial, scientific, planning or organising. But the cost is very high for something which may not be in constant intensive use — over £200, plus the cost of a disc system.

IT CAN'T BE DONE . . .

The story goes that the writers of VisiCalc were told by their mainframe colleagues that what they proposed to create was impossible, even on a large computer; today a spreadsheet for the ZX81 costs less than £10. In truth, most necessary routines, and certainly the most difficult ones to write, are part of the BASIC interpreter with which virtually every personal micro comes fitted. This is easily demonstrated by examining the main functional units required by a spreadsheet type of program.

A spreadsheet is in effect a generalised scheme for storing and manipulating labelled values. What facilities does BASIC offer in this respect?

Storing numeric or string values in known positions in a matrix of addresses is, of course, the way memory is used inside most computers. Higher level languages refer to the locations in which data is held by means of symbolic addresses known as variables, and most BASICs provide for a fairly large number of these. A BASIC interpreter will, therefore, have routines which store the assigned value of a variable together with its name in some place in memory where other routines can reliably find it again whenever necessary. And, most BASICs will accept variables named, like the cells of a spreadsheet matrix, with either a letter and a number or with two letters.

A floating point numeric value is stored by BASIC in 32 bits expressed as a power of two — like a kind of binary logarithm. The contents of a string variable are stored in a similar way and usually in the same general area as the numeric kind. They occupy only 24 bits; eight bits defining the length of the string and 16 forming the address pointer to the actual position at which the string is to be found. BASIC can also store array variables, using the same format as the simple kind but within an elaborate tabular structure for ease and speed of retrieval. So, BASIC is well equipped to service data consisting of simple numbers or strings.

STORE AND EVALUATE

The other need of a spreadsheet is the storage and evaluation of arithmetic expressions in the correct order. These are barely different in concept and not at all different in practice from the lines of instructions which make up so much of a BASIC program. BASIC requires that they be numbered and stored in numerical order; the lines will then be processed in that order unless otherwise instructed. Although the cells of the spreadsheet are normally not thought of as numbered, the two-character labels can indeed be regarded as numbers to base 26 and thus the equivalent of BASIC line numbers. The lines themselves are usually stored in compressed form, single bytes replacing the key words of the language as the arithmetic operators. This leads not only to much shorter stored programs but also ones which can be executed rapidly, since the process of decoding the instructions is thereby much simplified. When a new line is entered, any previous line of that number is first deleted and the new line is inserted in the correct place. When the program is run, each line is executed in turn and the interpreter converts the expressions from ASCII into whichever form is appropriate, performs the operations required and stores away the results by assigning new values to the variables involved. The lines themselves remain unchanged.

While the data storage arrangements of a program like VisiCalc share much in common with any BASIC program, what characterises this kind of package is its form of screen presentation. It needs to be able to seek out the value of all relevant variables and print them on the screen; it must also be able to display any stored expressions. BASIC too uses routines for PRINTing the values of variables and the LIST command converts the stored lines of code back from the tokenised form into full text accompanied by line numbers.

Thus, it seems that virtually all the fundamental needs of a spreadsheet are available somewhere within a BASIC interpreter. In fact, the whole program could be written in BASIC although it would be extremely slow and bulky. So what is now required is machine level software to harness the routines to perform tasks rather different from the ones they were originally designed to serve. But as a subroutine will perform its particular function irrespective of ultimate purpose (as long as the proper parameters are supplied to it on entry) this should not — and in the event does not — cause any problem.

As always, the most common difficulty encountered is that of trying to call a particular limited section of a program stored in ROM, especially if the ROM-based code has not been written in strictly modular form. Some interpreters make the task easier by being composed mainly of subroutines, with a convenient return at the end of each section, just as many monitors or operating systems make their routines available for general use. In this case, one simply CALLs the routine at its beginning and lets the RETurn at the end send control back to the calling program.

THE RIGHT TO CHOOSE

What we are looking for, of course, is an instruction which we can use to jump to an address of ***our*** choosing. And this can be done in other ways too. Occasionally, one of the lesser used registers arrives holding an address to which a jump like JP (IY) is made at the end of the sequence. Or, a return address is saved in a known location in RAM. But most frequently, while it is easy to decide at which address the call should begin, it is considerably less easy to get out again. All too often the desired function has long been performed, but there is no way to leave the ROM before half a dozen other less desirable, or even positively harmful, actions have taken place. The only way to cope with this is to copy the offending section of the interpreter into your own code, changing jump addresses where necessary. With luck the series of instructions will not be too long and, after all, you don't have to understand how it works.

Using an interpreter in this way depends greatly on setting the processor up with the right values in its registers on entry into the routine. Certain functions, particularly those processing text of some sort, require the 16-bit ▶

pointer registers, HL and DE, and sometimes the index registers, IX and IY, to contain pointers to the relevant position in the material. It is usually a relatively simple matter to follow the routine and note what data it uses . But care is required here. Sometimes what seems to be an irrelevant byte held in an unlikely place may seem to have no function in the matter in hand, but acts as a time bomb. Wilfully alter it, and the program suddenly and mysteriously crashes minutes later. Setting up the workspace correctly can either be done by inserting values in the relevant addresses or by the cruder, but no less effective, means of cold starting BASIC at some time before first calling its routines.

So how, and in what order, shall they be called? What, in short, is the algorithm to be? There are two possible ways of manipulating these figures. The window onto the memory displays a group of cells from the overall matrix. The values contained in these cells may very well depend on the values of other cells, some of them not displayed in the window. So, one approach would be to look at each cell in turn and calculate its value by moving in a controlled way up the tree of other related values.

Thus, for the expression:

G8 = F9 * 2

F9 is first sought out and found to be:

F9 = D1 + A2

D1 turns out to be:

D1 = A1 + E2/6

so A1 is looked for, evaluated in turn, and then E2. In this way, all the branches of the tree of definitions which comprise G8 are explored and the same is done for every cell before printing its content on the screen.

But handling this data structure is not a simple task unless a logic based high-level language like PROLOG is used (a PROLOG spreadsheet has in fact been written). Much simpler is to calculate the value of every cell of the matrix from beginning to end, before displaying the content of only a limited number; to do this, if all expressions are stored in order as lines of BASIC, requires no more than RUN.

Using the RUN routine has, however, an unexpected disadvantage. If all cell values are stored as BASIC variables, calling RUN will set them to zero unless they are protected in some way. The workspace of every BASIC interpreter holds pointers to the start and finish of the area of RAM in which it stores its variables. It is possible, by changing these pointers at appropriate times, to manage the memory and to ensure that values are not reset unintentionally. But this leads to much increased complexity. A far simpler method is to enter all cell content assignments, whether they are numbers, strings or expressions, as lines of BASIC. In this way, even though the variables disappear on calling RUN, they are immediately reinstated.

What this does is to create, out of sight of the user, a standard BASIC program — shades of the ' The Last One '! And it has an added advantage. It is often necessary to store the contents of the complete matrix for later examination or the addition of further values. A matrix in the form of a BASIC program can use BASIC's tape or disc storage routines to SAVE and LOAD in the usual way.

So the overall program need be no more complex than:

1. Display the contents of the visible cells.
2. Get an instruction from the keyboard and if appropriate, store it using the BASIC editor.
3. Run BASIC.
4. Go back to 1.

FOR EXAMPLE . . .

Applying these ideas to a real spreadsheet for my NASCOM, using Xtal's BASIC interpreter, has led to a package which I have called Mini-Calc. To examine the workings of the four steps in greater detail, it is best to look at this practical example.

Suppose the program is in use and we examine it after a number of times round the main loop. What do we see on the screen? A display of the matrix with an arrow pointing to one particular cell, offering us the opportunity to alter its content.

The Mini-Calc window displays 5 by 5 cells of the matrix. As the NASCOM has no reverse video to highlight the active cell, an up arrow is used as a cursor, which can be moved around the window to alter any cell displayed. Should the user try to move the arrow off the screen, the whole window will move to compensate. Thus, window movements are extensions of cursor movements. But, because the two are to some extent independant, Mini-Calc has to maintain a record both of which cells appear on the screen and to which of them the cursor points. In practice, the first of these is recorded as the name of the cell in the centre of the screen, from which the routine counts back two rows and two columns to give the cell in the top left-hand corner. Thereafter, the contents of each cell are printed using BASIC's print-variable routine, with the location which tells it where to print being primed from Mini-Calc's table of screen addresses.

Printing out the variable does require some care, however. BASIC has a routine to find whether a given variable exists but in many implementations of the language, if it is not found , it is created — with a value of zero or a null string. The effect of this would be to fill up the memory to no purpose and slow everything down. Mini-Calc, therefore, uses its own routine to search BASIC's variable storage area. This starts by holding the sought-after variable in one of the processor's register pairs and tries to match it against each variable name. But as initially it is not known whether the cell stores a number or a string and BASIC differentiates between the two by setting the top bit of a string variable's name, each variable is matched, first with bit 7 reset and then if that fails, with the top bit set. The routine returns with a flag set to show that the variable was not found, or else with the variable name stored in another two registers. To print its value, the variable name is deposited in BASIC's input buffer (followed by a '$' if found to be a string), a register pair is pointed at it and either the BASIC string or number printing routine is called.

This process would be sufficient by itself if the cells were only to contain values assigned to them directly by the user. But in many cases, the figures are the result of evaluating arithmetical expressions stored under the cell's name, and it is obviously necessary for the display to make this clear. But there isn't room on the screen for 25 expressions as well as the numbers or strings in the cells. So, of necessity, the only expression actually printed out is that which controls the value in the cell to which the cursor points and this is displayed on the bottom two lines.

The relevant line is found by converting the cell label to the equivalent line number and using a BASIC routine to search for that line. If it exists, LIST is used to de-tokenise and print it, otherwise a new prompt is issued, consisting of the cell name followed by an equals sign:

AA =

The program is now waiting for input from the user.

The input routine has to differentiate between four different kinds of instruction: a change of the cells included in the window , the assignment of a numeric value to the cell pointed at by the cursor, a string assignment to the cell, and the definition of the content of the cell in terms of an expression.

CONTROL THAT CURSOR

Mini-Calc makes use of the four cursor-control keys available on the NASCOM — Up, Down, Right and Left — to move the window and cell cursor around the memory. On other computers, the equivalent control key codes would be a suitable alternative.

If the command received is cursor up, the double movement of window and cursor is achieved in the following way. The cursor position is recorded as the name of the cell to which the cursor points and the screen address at which the up arrow is printed. The window position is recorded as the name of the cell at centre screen.

```
Decrement the window row twice to find top
screen row.

COMPARE it with the cursor row.
IF      they are not the same,
        DECREMENT the cursor row,
        STORE the new cursor cell,
        MOVE the arrow screen address,
        RESTORE the window row.
        END

ELSE
        IF top screen row is A, then END.

ELSE
        RESTORE window centre row,
        DECREMENT it,
        STORE it,
        DECREMENT cursor row,
        STORE it,
        (leave the arrow screen address as it
        is)
        END

END
```

Apart from cursor control, the three possible input assignments produce BASIC instructions of the form:

F8 = 234.5

or:

F8$ = "HELLO"

or:

F8 = +D1 +(B3/23.4)

It is convenient to differentiate between numeric and string assignments, so that the user is relieved of the need to insert the string marker, '$', and the quotation marks. It is no problem for the input routine to test which kind of assignment is required. This can be done by checking to see whether or not the input line consists only of numbers (in this context the point, '.', counts as numeric, else only whole numbers would be recognised). For the input routine to separate strings from expressions, Mini-Calc's syntax demands that expressions be preceded by a plus sign,'+', chosen because that sign has no effect on the value of the expression, and does not generate an error in the BASIC expression evaluation routine:

F8 = + D1 + (B3/23.4)

This evaluates to the same as:

F8 = D1 + (B3/23.4)

So far, for convenience, the cells have been described by a letter for the row and a number for the column — B2, F8 or K9 — but this may not be the best way to label them. Xtal BASIC, like many others, will support variables named AA to ZZ as well as A1 to Z9. Clearly the first option is preferable as it allows us to use 26 by 26 cells as opposed to only 26 by 10 for the second. 676 (ie 26 by 26) cells is, of course, not as many as in commercial spreadsheets — even the one for the ZX81 has 38 by 38 — but the number is sufficient to do quite a lot of useful work, and two-letter names are easy to translate to line numbers. Once the system is understood, there is no reason why a switch cannot be made to array variables, giving a matrix of 676 by n, ie AA (1) to ZZ (as-many-as-you-like), as long as one is prepared for the string manipulations needed to convert the array variable names to numbers.

So the prompt at the bottom of the screen is of the form:

BA =

and unless the user inputs a code to move the window, the input line will be printed to the screen following the prompt:

BA = 234

or:

BA = HELLO

or:

BA = + AA * 3

To pass this line to the BASIC editor, it only needs to be copied into the BASIC input buffer following the line number. Should the assignment be a string, the string marker,'$', and quotes are inserted during the copying process. Hence:

AA = HELLO

becomes:

AA$ = "HELLO

Notice that the closing quotes are *not* needed in this implementation of BASIC as long as the string ends the line.

Stage 1 of the overall program now becomes:

PRINT A PROMPT.
GET AN INPUT.

IF	the input is a cursor control code, jump to window movement routine.
ELSE	Turn variable name into a line number, AA ⟶ 26 Write the number into the beginning of the BASIC input buffer. 26 Copy the prompt and the line into the BASIC input buffer after the line number. 26 AA = 123.4 IF the assignment is a string THEN insert string marker and quotes. 26 AA$ = "HELLO

CALL BASIC EDITOR.

This is all that a simple spreadsheet needs for its input stage, but there are two possible extensions which would make it much easier to use in practice. One is 'expansion' and the other is 'replication' and it is at this stage, before the BASIC editor is called, that 'expansion' is carried out.

EXPANDING THE FACILITIES

Suppose one wishes to sum all or most of the cells in one particular column or row. This would be expressed as follows:

JA = + AA+ BA+ CA+ DA+ EA+ FA+ GA+ HA+ IA

But what a chore to have to type all that in, and it is only a third of the full column! Mini-Calc allows the user to enter the same expression in the form:

JA = + [AA+ IA]

The part of the line between square brackets is expanded to include all of the cells between those named, and the stated arithmetic operator is applied between them all. In the example given, it is '+', but it could equally be '−', '*', '∧' or even '/'.

The expansion is performed at the time the assignment is copied into the BASIC's buffer; the first three characters (AA +) are held in three of the processor's registers and repeatedly re-copied, incrementing either the row or column part of the label (BA+, CA+, DA+, etc), until it is the same as the last two characters between the brackets (IA). In this way BASIC receives and stores the line in its full form.

Once the line is safely in the buffer, BASIC is called to store it away in its proper place in the sequence of lines before returning control to Mini-Calc. At this point, either 'RUN' is called, or else the other additional facility: 'replication'.

This constitutes a short cut which allows the user to enter expressions into a number of cells all at the same time. Suppose column B contained the prices of a range of goods stocked by a computer store — computers in BB, disc drives in CB, VDUs in DB and so on, ranging up to cell FB — and column C holds the numbers sold in a given month. The store owner wishes to calculate the monthly gross returns for each item in column D.

The expressions he or she would have to type in are all of the same form. The only difference is that as the row label of the cell in which he or she is placing the expression increases so do the row labels of the other cells involved.

	A	B	C	D
A	ITEM	COST	SOLD	GROSS
B	computers	500	5	BD=+BB*BC
C	drives	400	8	CD=+CB*CC
D	VDUs	200	6	DD=+DB*DC
E	printers	300	7	ED=+EB*EC
F	joysticks	60	10	FD=+FB*FC

Most spreadsheets provide a means to do this automatically and Mini-Calc is no different. In the case just shown, the user would enter the following in cell:

BD = [FD=] + BB * BC

and the software would input the relevant expressions into all the cells from BD to FD, updating BB and BC row by row to FB and FC as it does so.

But the adjustment is not just performed on rows or columns with the same label as the target cell. In fact, as this is incremented, so will be the row or column of all cells involved, so that calculating a continuous multiplication such as compound interest is easy. Start with a number in AA and then enter:

BA = [MA =] + AA * 1.5.

MA will then show the result after a year of 20% per month interest on the sum in AA.

Mini-Calc does this by setting a flag when the input routine recognises the replication request. After the BASIC editor has entered the line, instead of calling the RUN routine right away, the same line, but with updated cell labels, is entered into the next cell and the BASIC editor is called again. This process is continued until the last cell assigned an expression is the one named between the square brackets and only after that, is the flag reset and RUN finally called.

THE FINAL PRODUCT

What then is the program that is RUN? It consists of all the variable assignments and expressions entered into the cells of the matrix, stored at line numbers derived from the cell labels — a possible 676 of them. But Mini-Calc actually starts out with a BASIC program even before anything is input. Xtal BASIC provides an ON ERROR statement which can be used to prevent an unexpected and unnerving crash into the BASIC editor environment. This is not essential, of course, although very useful. But one line at least must be there: the final one which sends control back to Stage 1 of the spreadsheet's main loop. This is a USR or CALL instruction, which fools the BASIC program into believing that it is simply calling a machine code utility. Once back in Mini-Calc, however, the return address is promptly popped off the stack and thrown away. This line is truly — The Last One!

HOW TO GET IT

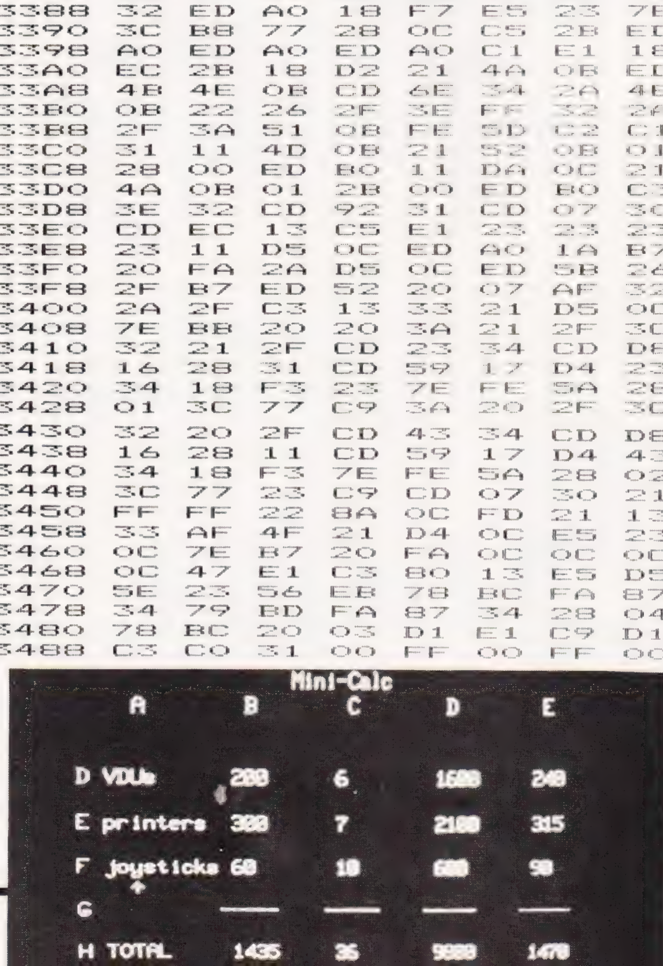
The following Hex dump is the machine code program that, together with Xtal BASIC 2.2, makes up the Mini-Calc package. Whilst this is immediately usable by NASCOM owners this might not suit your needs. In next month's issue we will be publishing some notes that relate to this article for users of 6502-based systems. A fully commented ZEAP assembler listing is available from our offices for £9.95 and the program on its own and as a special package with Xtal BASIC will shortly be available through CT Software.

If you are sufficiently convinced that your version of this program, for whatever system, is as good as our original, please contact us as we are very interested in producing other versions under the CT Software label.

Xtal BASIC is available from Crystal Electronics of Torquay in a number of forms for a variety of machines.

Program Listing

```
2F20 41 41 43 43 0E 09 41 47
2F28 41 41 00 00 01 00 05 11
2F30 76 2F 3A 23 2F D6 02 EB
2F38 5E 23 56 23 EB FE 41 FA
2F40 68 2F FE 5B F2 68 2F 77
2F48 3C 10 EC 01 00 05 3A 22
2F50 2F D6 02 EB 5E 23 56 23
2F58 EB FE 41 FA 6F 2F FE 5B
2F60 F2 6F 2F 77 3C 10 EC C9
2F68 F5 3E 20 77 F1 18 D9 F5
2F70 3E 20 77 F1 18 EE 0E 08
2F78 17 08 20 08 29 08 32 08
2F80 CA 08 4A 09 CA 09 4A 0A
2F88 CA 0A 2A B7 0C E5 ED 5B
2F90 B9 0C A7 ED 52 E1 C8 56
2F98 23 5E E5 2A 20 2F A7 ED
2FA0 52 28 11 2A 20 2F CB FC
2FA8 A7 ED 52 28 07 11 05 00
2FB0 E1 19 18 D9 E1 37 C9 22
2FB8 29 0C CD 8A 2F 30 17 CB
2FC0 7A 20 1E ED 53 D5 0C E5
2FC8 21 D5 0C CD 7A 31 CD B2
2FD0 19 CD FB 2F E1 C9 EF 20
2FD8 20 20 20 20 20 20 20 00
2FE0 C9 CB BA ED 53 D5 0C 3E
2FE8 24 32 D7 0C CD 7A 31 E5
2FF0 21 D5 0C CD B2 19 CD FB
2FF8 2F E1 C9 E5 21 00 00 22
3000 D5 0C 22 D7 0C E1 C9 3A
3008 21 2F D6 40 47 21 00 00
3010 11 1A 00 19 10 FD 3A 20
3018 2F D6 41 5F AF 57 19 EB
3020 C9 EB FD 21 51 30 D9 E5
3028 21 D5 0C D9 AF FD 5E 00
3030 FD 56 01 B7 ED 52 38 03
3038 3C 18 F8 19 C6 30 D9 77
3040 23 D9 D6 30 FD 23 FD 23
3048 7B FE 01 20 DF D9 E1 D9
3050 C9 10 27 E8 03 64 00 0A
3058 00 01 00 3A 20 2F 21 22
3060 2F 35 35 BE F5 34 34 F1
3068 28 0E 2A 24 2F 11 80 00
3070 B7 ED 52 22 24 2F 18 05
3078 FE 41 28 05 35 3D 32 20
3080 2F C3 C1 31 3A 20 2F 21
3088 22 2F 34 34 BE F5 35 35
3090 F1 28 0C 2A 24 2F 11 80
3098 00 19 22 24 2F 18 05 FE
30A0 5A 28 05 34 3C 32 20 2F
30A8 C3 C1 31 3A 21 2F 21 23
30B0 2F 35 35 BE F5 34 34 F1
30B8 28 0E 2A 24 2F 11 09 00
30C0 B7 ED 52 22 24 2F 18 05
30C8 FE 41 28 05 35 3D 32 21
30D0 2F C3 C1 31 3A 21 2F 21
30D8 23 2F 34 34 BE F5 35 35
30E0 F1 28 0C 2A 24 2F 11 09
30E8 00 19 22 24 2F 18 05 FE
30F0 5A 28 05 34 3C 32 21 2F
30F8 C3 C1 31 C5 E1 23 23 23
3100 23 E5 23 23 23 7E FE A8
3108 E1 C2 D6 31 CD CF 15 C3
3110 E1 31 CD 81 31 2A 20 2F
3118 22 28 2F CD 2C 2F CD FB
3120 2F 2A 20 2F E5 2A 22 2F
3128 22 20 2F 7D D6 03 6F 7C
3130 C6 03 67 22 20 2F 21 80
3138 2F 06 05 ED 5B 20 2F 7A
3140 D6 05 57 1C ED 53 20 2F
3148 5E 23 56 23 13 13 E5 EB
3150 C5 06 05 E5 C5 CD B7 2F
3158 C1 ED 5B 20 2F 14 ED 53
3160 20 2F 11 09 00 E1 19 10
3168 EA C1 E1 10 CE E1 22 20
3170 2F F5 3E 5E 2A 24 2F 77
3178 F1 C9 C5 47 AF 3C 78 C1
3180 C9 21 0A 08 22 29 0C 06
3188 0F C5 EF 1B 0D 00 C1 10
3190 F8 C9 E5 C5 F5 21 D5 0C
3198 01 60 00 3E 20 77 10 FD
31A0 F1 C1 E1 C9 CD 07 30 CD
31A8 EC 13 30 15 21 8A 0B 22
31B0 29 0C 3E 1B F7 18 50 FE
31B8 2E C8 3E 01 32 90 0C C9
31C0 E1 CD 12 31 21 4A 0B 22
31C8 29 0C CD 92 31 CD 07 30
31D0 CD EC 13 DA FB 30 2A 20
31D8 2F 22 4A 0B 3E 3D 32 4C
31E0 0B 21 4D 0B 22 29 0C AF
31E8 DF 7B F7 FE 11 CA AB 30
31F0 FE 12 CA D4 30 FE 13 CA
31F8 5B 30 FE 14 CA 84 30 FE
3200 0D 28 BE FE 1B 28 9D CD
3208 07 30 CD 21 30 DF 7B FE
3210 0D 28 07 FE 1B 28 8D F7
3218 18 F3 AF 32 2A 2F 3A 4D
3220 0B FE 5B CA A4 33 01 2B
3228 00 21 4E 0B 3E 5B ED B1
3230 CA 40 33 11 DA 0C 21 4A
3238 0B 01 30 00 ED B0 21 8A
3240 0B 01 2B 00 ED B0 1B 1A
3248 FE 20 20 03 1B 18 F8 13
3250 C1 AF 12 3A 4D 0B FE 2B
3258 28 35 AF 32 90 0C 21 DD
3260 0C CD D8 16 28 05 D4 B7
3268 31 18 F6 3A 90 0C B7 28
3270 1E 21 E6 0C 11 E8 0C 01
3278 0A 00 ED B8 11 24 3D ED
3280 53 DC 0C 3E 22 32 DE 0C
3288 32 E9 0C AF 32 EA 0C 21
3290 FF FF 22 8A 0C FD 21 13
3298 33 B7 21 D6 0C CD D8 16
32A0 CA 5C 13 F5 CD 8C 17 D5
32A8 CD 49 14 F5 C5 DD E1 D5
32B0 E5 0D 0D 0D 23 06 00 7E
32B8 FE 8B CC E3 32 FE 93 CC
32C0 E8 32 FE A2 CC ED 32 FE
32C8 A3 CC F2 32 FE AE CC F7
32D0 32 FE CD CC FC 32 23 0D
32D8 20 DD E1 D1 DD E5 C1 F1
32E0 C3 7A 13 11 49 46 18 17
32E8 11 4F 4E 18 12 11 54 4F
32F0 18 0D 11 46 4E 18 08 11
32F8 4F 52 18 03 11 50 49 C5
3300 D5 E5 09 54 5D 1C ED B8
3308 E1 D1 73 23 72 C1 0C DD
3310 23 AF C9 3A 2A 2F B7 C2
3318 DA 33 2A 28 2F 22 20 2F
3320 21 FF FF FD 21 C0 31 22
3328 8A 0C 3E 8A 32 D5 0C AF
3330 32 D6 0C 47 0E 05 21 D4
3338 0C 11 00 00 B7 C3 B0 16
3340 11 DA 0C 21 4A 0B 01 2B
3348 00 7E FE 5B 28 04 ED A0
3350 18 F7 23 E5 ED A0 ED A0
3358 ED A0 4E 23 46 E1 CD 6E
3360 34 7E B9 28 28 E5 3C B9
3368 77 28 0B C5 ED A0 ED A0
3370 ED A0 C1 E1 18 EF ED A0
3378 ED A0 23 23 23 7E FE 5D
3380 C2 C1 31 23 7E B7 CA 46
3388 32 ED A0 18 F7 E5 23 7E
3390 3C B8 77 28 0C C5 2B ED
3398 A0 ED A0 ED A0 C1 E1 18
33A0 EC 2B 18 D2 21 4A 0B ED
33A8 4B 4E 0B CD 6E 34 2A 4E
33B0 0B 22 26 2F 3E FF 32 2A
33B8 2F 3A 51 0B FE 5D C2 C1
33C0 31 11 4D 0B 21 52 0B 01
33C8 28 00 ED B0 11 DA 0C 21
33D0 4A 0B 01 2B 00 ED B0 C3
33D8 3E 32 CD 92 31 CD 07 30
33E0 CD EC 13 C5 E1 23 23 23
33E8 23 11 D5 0C ED A0 1A B7
33F0 20 FA 2A D5 0C ED 5B 26
33F8 2F B7 ED 52 20 07 AF 32
3400 2A 2F C3 13 33 21 D5 0C
3408 7E BB 20 20 3A 21 2F 3C
3410 32 21 2F CD 23 34 CD D8
3418 16 28 31 CD 59 17 D4 23
3420 34 18 F3 23 7E FE 5A 28
3428 01 3C 77 C9 3A 20 2F 3C
3430 32 20 2F CD 43 34 CD D8
3438 16 28 11 CD 59 17 D4 43
3440 34 18 F3 7E FE 5A 28 02
3448 3C 77 23 C9 CD 07 30 21
3450 FF FF 22 8A 0C FD 21 13
3458 33 AF 4F 21 D4 0C E5 23
3460 0C 7E B7 20 FA 0C 0C 0C
3468 0C 47 E1 C3 80 13 E5 D5
3470 5E 23 56 EB 78 BC FA 87
3478 34 79 BD FA 87 34 28 04
3480 78 BC 20 03 D1 E1 C9 D1
3488 C3 C0 31 00 FF 00 FF 00
```



COMPETITION

Not satisfied with giving away a complete BASIC tutorial book over the next three months, we decided it was time to be really generous. Looking around the marketplace to see what we could offer, we realised that it just had to be a ZX Spectrum and, because we're not at all mean, we thought that we'd add a few extras too . . . no, not a Microdrive!

First prize
A 48K ZX Spectrum complete with ZX Printer and £25 worth of Spectrum software

Second prize
A ZX Printer

Third prize
£25 worth of Spectrum software

The competition could hardly be simpler — we don't like difficult competitions as they're too hard to mark. All you have to do is to find the 15 words directly relating to Sinclair and the ZX Spectrum which are buried in the Wordsquare. A helpful hint: There are a number of dummy words but all the real ones can be found by looking through the August '82 issue of *Computing Today*. Well, that's not quite all. A number of the hidden words share letters with other words and we'd like you to make a note of these shared letters. Now, fill in the blanks to prove that you've found the words, circle the shared letters and write your name and address on the coupon.

Simple . . . so far! Cut the coupon out of the magazine, complete with the Wordsquare, and stick it in an envelope. Now, before you post it, add up the values of the shared letters, A = 1 . . . Z = 26, convert the total to hexadecimal and write this on the back of the envelope!

Now you can post it to us.

CT Spectrum Competition

```
U O W D S F F W E O Q Y T P A _ _ _ _ _ _ _ _ _ _
Y Y P B D I U B V Q Y Z W T J _ _ _ _ _ _ _ _ _
J K Y U R I I J I T E M F S T _ _ _ _ _ _ _ _
E L E Q E T E G R S L B J D J _ _ _ _ _ _ _ _
O O K U E N R I D R R F A E T _ _ _ _ _ _ _
L O G J B A O O O E E F H L H _ _ _ _ _ _ _
Z U U B D R T H R K B P Y R G _ _ _ _ _ _
T A V L A R S K C C M I D A I _ _ _ _ _ _
Y L R H R E E A I I A S K E R _ _ _ _ _ _
P U V T B F R L M V C Q A S B _ _ _ _ _ _
A G L I I A P P E I E O E Y L _ _ _ _ _
P K N B M M J N R B Y Z F V M _ _ _ _ _
E L D T E T E C G L N W E H A _ _ _ _ _
R S O J U N L X E H N L Q X X _ _ _
J P V G W E J X J D S Y N R S _ _ _
```

NAME ....

ADDRESS ....

....

....

.... **Postcode** ....

RULES

This competition is open to all UK and Northern Ireland readers of Computing Today except employees of Argus Specialist Publications Ltd, their printers, distributors, employees of Sinclair Research Limited and their distributors or anyone associated with the competition.

All entries must be submitted on the coupon cut from the magazine — photocopies will not be accepted.

As long as the correct coupon is used for each entry, there is no limit to the number of entries.

All entries must have the code number written on the outer flap of the envelope, entries without this number will not be accepted.

All entries must be postmarked before 31st October 1982.

The prizes will be awarded to the first three correct entries drawn after the closing date.

No correspondence will be entered into with regard to the results and it is a condition of entry that the Editor's decision is accepted as final.

The winners will be notified by post and the results will be published in a future issue of Computing Today.

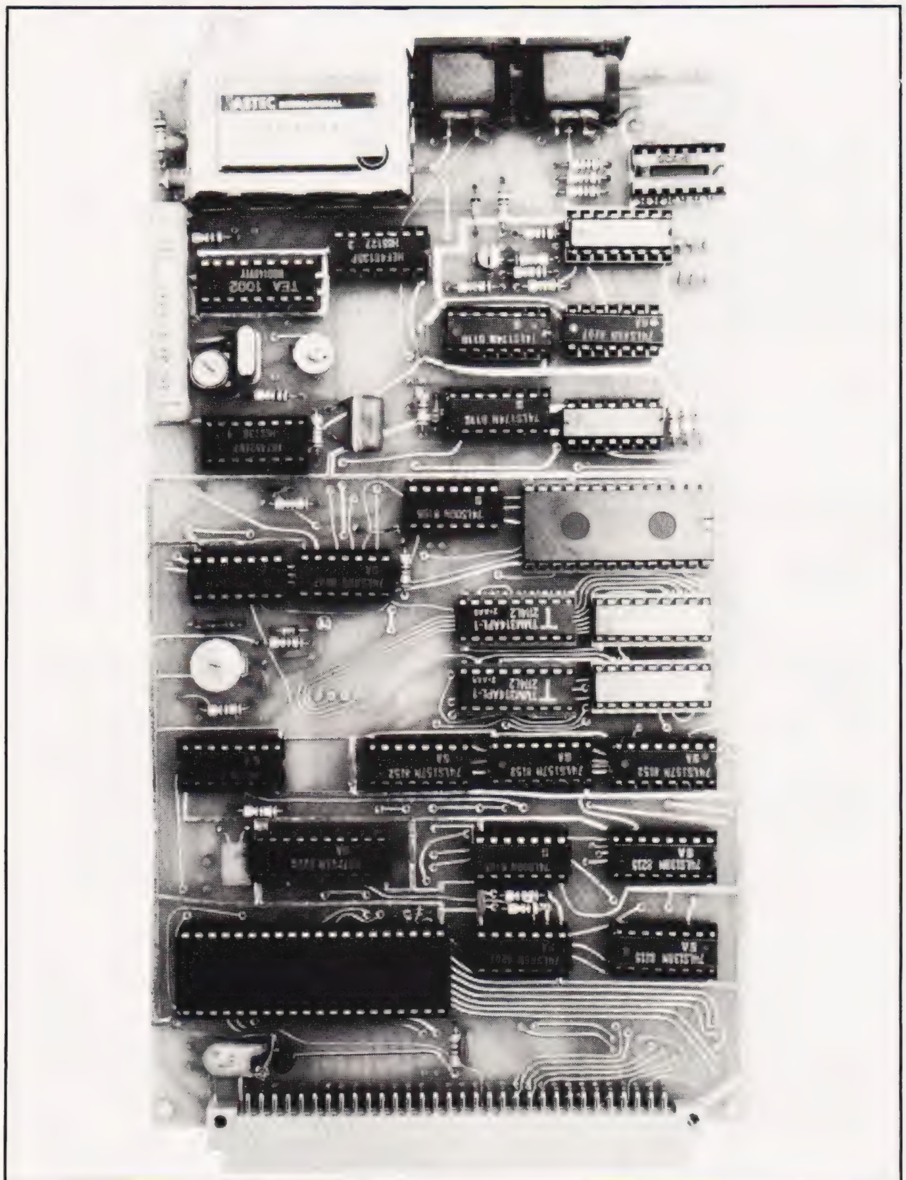
Address your answers to:

COMPUTING TODAY
Competition,
145 Charing Cross Road,
London WC2H 0EE.

Ian Graham

SPECIAL REPORT

Overcoming Tangerine's limited display with an add-on.



The Microtan 65 first saw the light of day in *Computing Today* in June 1980. Designed and built by the all-British outfit, Tangerine Computers, this 6502-based system can be built up board by board into as sophisticated a personal computer system as you need (or can afford).

In its simplest form it is a power supply, CPU board, Hex keypad and a television set, allowing machine code or 6502 assembly language programming. Add a second board and you have BASIC, lower case, graphics and 7K of RAM to play with. The screen format is 16 lines of 32 characters and while this is adequate for simple programming applications, greater flexibility is offered by a 64 column by 25 row display in terms of both the length of listing which can be displayed and the smaller size of characters available for graphics.

Although Tangerine Computers have expanded the original two-board system so that it now includes a full ASCII keyboard, a 40K RAM board, a disc operating system and a full system rack with switch-mode power supply, they have never updated the 32 by 16 screen format. A design team which goes by the unlikely name of Mousepacket Designs have filled the gap left by Tangerine with a plug-in board giving the system a 64 by 25 colour display.

The Mousepacket board plugs directly into the system rack, in any of the numbered sockets. In addition, two of Tangerine's chips must be replaced by new chips supplied by Mousepacket Designs. One replaces Tanbug, Microtan's monitor program located on the Microtan 65 board, and the other replaces the top 2K of the BASIC interpreter in position D3 on Tanex. No other adjustments need be made. Power up the computer and connect a monitor or TV to Mousepacket's board. The screen should be full of characters — letters, numbers, reverse characters and graphics, some flashing. If the screen is blank, pressing Reset should elicit a response. TANBUG is now printed in the bottom left corner of the screen, with a flashing underline cursor below it.

Because all the original Tanbug routines are preserved, no special commands are necessary. Typing **BAS** brings the familiar responses MEMORY SIZE? and TERMINAL WIDTH? followed by the amount of the available user RAM.

COMPATIBILITY CARED FOR

BASIC programs written for the old screen (32 by 16) will run on the new screen providing information is output by means of PRINT statements. The old screen information will, of course, occupy only half the line length of the new screen. Programs which POKE data directly into the screen memory will need to be modified.

The old screen display memory occupies locations 200 to 3FF Hex with the cursor residing at 3E0 Hex which is the bottom left-hand corner of the screen. The new screen occupies memory space from B000 to B63F Hex and any other boards added to the system must be configured to occupy space below B000 Hex. Also, there must be a discontinuity between the user RAM and the screen memory. That is easily achieved by removing the chips in positions B1 and B2 on TANRAM. BASIC uses the top of the available memory for string storage and without the discontinuity, it would try to use the screen memory.

Three additional locations on the stack, 150 to 152 Hex, specify the current line length, cursor start and screen change flag. For most applications, they can be left to look

FACTSHEET

Format Mousepacket 64 by 25 colour VDU
25 lines of 64 characters
128 by 75 pixel capability
Full Teletext character set
Full colour selection
Monitor or TV outputs

Cost Approximately £70

Supplier Mousepacket Designs
7 Cedar Close
Grafham
Huntingdon
PE18 0DZ
0480-810550
(Also available through Microtanic)

after themselves. However, ambitious programmers can use them to advantage. The operating conditions of the CRT controller on the VDU board are set by a number of registers. Five of these, numbered 10 to 15, are available to the user to control the height, position and blink rate of the cursor and the screen display status.

Register number 10, for instance, determines the cursor's top line number and display status. Its contents are stored in location 151 Hex in the stack. By modifying register 10, the user can select a fast or slow cursor blink rate, a non-blinking cursor or a non-displayed cursor. Registers 12 and 13 determine the screen start address. In practice, this means that by modifying the contents of registers 12 and 13, the user can make the screen scroll up, down, left or right from within a program.

RAM FOR FREE?

It may have struck you that if a new screen memory space is being used by Mousepacket's board, the old Microtan screen memory is lying around unused and available as user RAM. That is broadly true, but there are two points to be made. The first 32 characters of every line output to the new screen are also output to the bottom line of the old screen; the tape I/O routines in XBUG and BASIC monitor these locations for user responses. The remainder of the old screen, 200 to 3DF Hex, is available as user RAM. As Mousepacket's manual points out, this is particularly useful for the storage of machine code routines because the old screen memory lies below the BASIC storage space. Note that on-screen editing in BASIC is fully preserved, but that this is performed on the old screen and then copied to the new screen. This inevitably means that both screen memory areas are corrupted and routines stored in them should be saved before editing.

The second point to note is that when the Mousepacket board is fitted, the old screen is not lost forever. It is possible to connect a monitor to the new board and a television set to the existing Microtan UHF modulator output and switch between the two either directly from the keyboard or under program control. This has the advantage that programs which run perfectly adequately on the old screen need not be modified — a relief if you've acquired a few dozen programs! However, if you plan to stick to the Mousepacket board and its screen format from now on, programs will have to written (and old programs re-written) to take the new format into account.

A GRAPHIC DISPLAY

For the purpose of this test, I used a 9" green screen monitor connected to the DIN socket output of the board although for best results a colour monitor should be used. Although I was unable to assess the system's performance in colour, I gave it a good run through using the monochrome monitor, working from the Mousepacket manual. The system uses the Teletext character set; the only surprise arising from that is encountered when programming in 6502 assembly language.

When loading data into a register, the accumulator for example, the computer determines whether the data is in binary, decimal or hexadecimal by the prefix keyed in. The prefix '# ' indicates Hex, but on typing this in, it is output to the screen as a '£'. However, it is still interpreted by the system as a prefix to Hex data. Otherwise, the Microtan XBUG translator and disassembler work in the normal way on the new screen, although the ASCII equivalents of the code are not displayed and the flashing cursor is not suppressed.

The most important point to note is that the Teletext character set offers enormous flexibility, especially in terms of graphics. Even textual information can be presented in a much more eye-catching manner. In addition to the freedom to specify the colours of background and characters, the characters can be made to flash or appear in reverse video by burying control codes in the program. Colours are shown on a monochrome monitor as different shades of grey. Contiguous graphics (no spaces between the lines) can also be selected, if desired. The Mousepacket design team stressed that results with a colour or monochrome television receiver can be disappointing and so a monitor should be used.

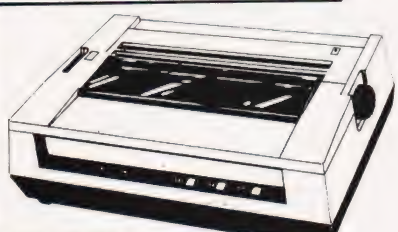
CONCLUSION

I was particularly excited by the idea of a 64 by 25 display format, because I use my Microtan rack system as a word processor with the assistance of the WP package written by John Dawson. The existing 32 by 16 format, however, is very restricting. The 64 by 25 format is more like that of the ubiquitous A4 sheet of paper, the office standard, so that the screen can be made to show a 'page' of text at a time. Unfortunately, the software package will not run with the new board. Here's hoping that Mr Dawson will recognise the value of the 64 by 25 format and come up with a revised WP package — I'll be the first customer.

This modification to the Microtan system is long overdue. At an expected selling price of £69.95, it represents reasonably good value for money — compare Beeline's 40-column board for the VIC at over £200! ■

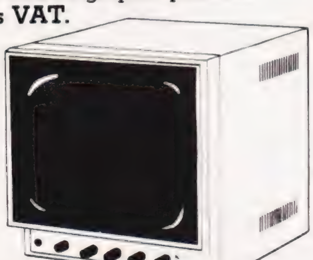
ingenious!

Genie I and II accessories



PRINTERS
The EG 3085 is quiet, fast and efficient. Printspeed is 100 characters per second and printing is bi-directional at 80 or 136 characters per line. Suitable for use with other systems, it has three typestyles, adjustable pin or friction feed and single sheet or roll paper facilities. **£425 plus VAT.**

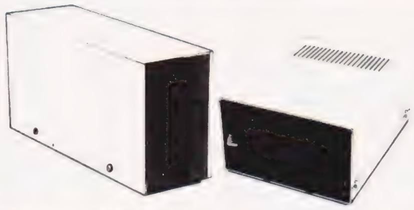
If you don't want to pay that much for a printer, consider the EG 603. It doesn't match the EG 3085 in certain areas, but you will still get 100 c.p.s. bi-directional, a range of character styles, forward and reverse feed and pin/friction feed, with 96 characters and 64 graphic patterns. **£235 plus VAT.**



MONITORS
Available in 9" and 12" sizes, with white, green or amber display, Lowe A.V.T. monitors are sturdy, attractive, easy to operate and feature an easy view screen with smoked anti-glare display filter. Compatible with most popular micros on the market. **From £75 plus VAT.**

EQUIPMENT COVERS
Beat the dirt, coffee spills and sticky fingers when your computer and monitor are not in use with these top quality black leather covers.

Genie CV1 **£5 plus VAT.**
12" monitors CV6 **£6.20 plus VAT.**
9" monitors CV3 **£4.20 plus VAT.**

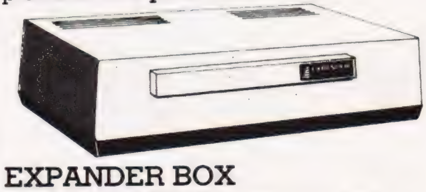


DISK DRIVES
If you want fast, reliable program storage, true random access file handling and access to many computer languages, we can meet your needs. The EG 400T provides storage of up to 184320 bytes per floppy disk and comes complete and tested, in a stylish colour matched cabinet. **£220 plus VAT.**

The EG 401 AT offers dual disk drive with 368640 bytes of useable storage and comes complete with a power supply ready to connect to an expander box. **£365 plus VAT.**

DOUBLE DENSITY ADAPTOR
Allows the use of standard minidisk drives in double density, with virtually double the storage capacity. The EG 3021 is equally at home in the Genie or TRS-80 expander boxes.

A double density disk operating system will be needed, such as smalLDOS provides. **£72 plus VAT.**



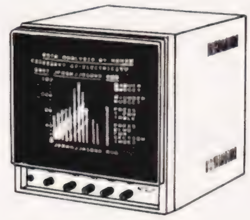
EXPANDER BOX
The updated EG 3014 expander box allows for up to four disk drives with optional double density. It connects to a printer, or RS 232 interface, or S100 cards. Not bad value at **£190 plus VAT (16K version) or £200 plus VAT (32K version)**

*The EG 3014 will work with TRS 80 by using the EG 3023 Tandy Adaptor.

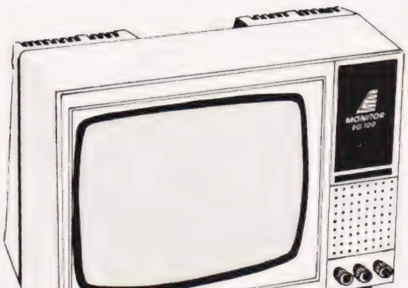
TECHNICAL MANUALS
Full technical details of Genie Hardware (all you ever wanted to know about Genie).

Genie I/II Technical Manual
£10 - No VAT.
Expander and accessories (EG3014)
£10 - No VAT.

smalLDOS
Powerful, yet reasonably priced, the Genie smalLDOS contains 21 library commands, 7 utilities, LBASIC, disk basic and bags of information, including a reference manual and 40 page beginners guide to disk usage. **£35 plus VAT.**



HIGH RESOLUTION GRAPHICS
Increase graphic resolution capabilities on your Genie seventy-three fold with the LE18 HI-RES unit. It offers bit image graphics of 73,728 points, a resolution of 384 x 192, and uses a separate 16K of video memory to achieve its resolution. Graphics are intermixable with text or existing pixtel graphics, and animation, reverse video displays and use of programmable graphic characters are possible. **£86 plus VAT.**



GENIE MONITORS
Two good performance, low priced 12" monitors, either to match your Genie or compatible with a wide range of other systems. Good resolution and band width and, of course, they free your television set for the other type of programmes you like to watch!

The EG 100 12" in black & white costs **£69 plus VAT.**
The EG 101 12" with green phospher is **£79 plus VAT.**

G P Davis

BUFFERED PROGRAMMING

How to get your program writing programs.

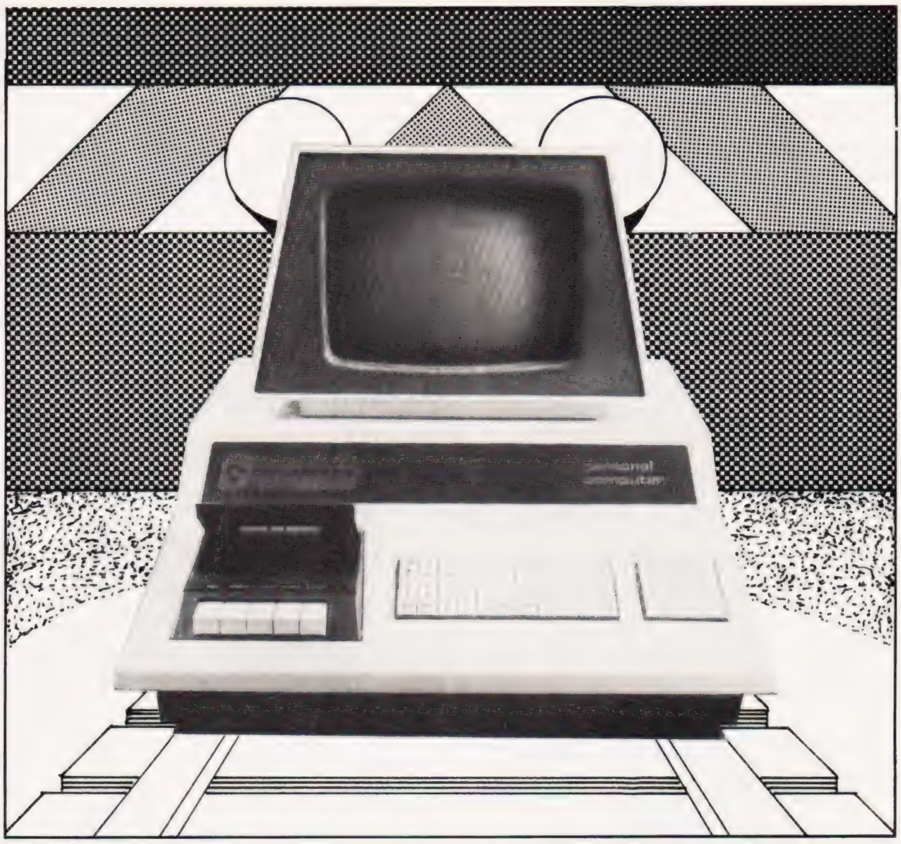
Most microcomputers use a keyboard buffer to provide temporary storage for a command or a line of BASIC program as it is being keyed in. When the input is terminated with a Carriage Return, the contents of the buffer are either processed and the command carried out or else the program line is transferred to its appropriate location in the computer memory. Although precise details of the mode of operation of the keyboard buffer vary for different machines (this article concentrates on the use of the PET keyboard buffer) the general principles described here are common to other computers.

When characters are keyed into the PET they are stored progressively from memory location 623 (decimal) onwards; at the same time, the number of characters is counted and stored in memory location 158 (decimal). On completion of a program, the PET examines the character count at location 158 and, if the result is non-zero, processes the number of characters specified in the character count location. If, for example, a line of BASIC has been loaded into the keyboard buffer by running a suitable program, then this line will be added to the original program as soon as the program is completed. This facility can be used to advantage in writing a number of useful 'toolkit'-type programs such as auto-numbering, REM removal, program contraction, etc.

CALL MY BUFF

The program listed below demonstrates the use of the keyboard buffer to give commands and to automatically write a BASIC program. As it is a 'toolkit' program and must exist in memory together with the program to be operated on, all REMs have been omitted in order to conserve space. The various steps involved are discussed in some detail below.

Apart from its use as an example of a keyboard buffer program, it will also convert a machine code routine into the DATA statement format as required for insertion in a BASIC program. Converting hexadecimal op-codes into decimal is rather laborious and the subsequent keying in of apparently meaningless lines of figures can lead to errors which are difficult to locate. This program eliminates both difficulties as it scans the hexadecimal code, converts it to decimal and finally writes and stores the converted



Program Listing

```
9000  INPUT "START LINE NUMBER";L
9010  INPUT "START, END OF CODE";S,E
9020  T1=INT(S/256):T2=INT(E/256)
9030  POKE 8184,T1:POKE 8185,S-T1*256
9040  POKE 8186,T2:POKE 8187,E-T2*256
9050  I=0:GOSUB 9220
9060  S=PEEK(8184)*256+PEEK(8185)
9070  E=PEEK(8186)*256+PEEK(8187)
9080  I=PEEK(8188)*256+PEEK(8189)
9090  L=PEEK(8190)*256+PEEK(8191)
9100  PRINT "[CLS]";L;"[^D][^A][^T][^A]";
9110  PRINT PEEK(S+I);
9120  I=I+1:IF S+I>E THEN 9200
9130  IF I/8=INT(I/8) THEN 9150
9140  PRINT ",";:GOTO 9110
9150  PRINT "X":L=L+10:GOSUB 9260
9160  POKE 624+K,71:POKE 625+K,79:POKE 626+K,84
9170  POKE 627+K,79:POKE 628+K,57:POKE 629+K,48
9180  POKE 630+K,54:POKE 631+K,48:POKE 632+K,13
9190  POKE 158,K+10:GOSUB 9220:END
9200  PRINT "Y":GOSUB 9260
9210  POKE 158,K+1:END
9220  T1=INT(L/256):T2=INT(I/256)
9230  POKE 8190,T1:POKE 8191,L-T1*256
9240  POKE 8188,T2:POKE 8189,I-T2*256
9250  RETURN
9260  V=32769:J=0:K=0
9270  B=PEEK(V+J)
9280  IF B=32 THEN J=J+1:GOTO 9270
9290  IF B<>24 AND B<>25 THEN 9310
9300  POKE 623+K,13:RETURN
9310  POKE 623+K,PEEK(V+J):K=K+1:J=J+1
9320  GOTO 9270
```

code into lines of BASIC in DATA statement format.

Before discussing the program in detail it should be noted that whenever a BASIC program is altered, the computer 'forgets' where it has stored all the variables. It is possible to write a further program to 'remind' the machine where it has stored the variables but when the number of variables is not too great, it is often easier to reserve fixed memory locations for the variables and to POKE and PEEK their values as required.

The four unusual symbols in line 9100 are necessary because of the mixed coding used in PET graphics. The symbols are entered by keying in 'shifted' DATA. When the graphic display locations of the symbols are PEEKed in line 9270, the correct ASCII codes for the letters DATA are stored in the buffer.

To use the program enter the machine code somewhere in memory as explained below and test the coding. Then enter the program and RUN 9000.

The program occupies some 780 bytes of memory and uses the keyboard buffer and part of the second cassette buffer. When the program is run it writes a new section of BASIC which requires some four bytes of memory for each byte of machine code. To prevent overwriting, the machine code should be started at an address later than 1805 plus four times the number of machine code bytes. The machine code must end before the first of the variable locations (8184).

Line	Function
9000-9010	Input the line number where the DATA statements are required to start and the start and finish addresses (decimal) of the machine code to be converted.
9020-9040 9220-9240	Store variables in locations 8184 to 8191 (ie the last eight bytes of 8K machine).
9060-9090	Recall variables for use in the program.
9100	Clears screen and displays line number (L) followed by four symbols (see below).
9110	Displays eight Hex codes converted into decimal.
9120	Tests for end of machine code.
9130	Ensures that only eight codes are displayed.
9140	Prints a comma after each DATA entry and gets the next code.
9150	Prints letter X at end of line and increments the line number counter.
9160-9190	Load GOTO 9060 (CR) into buffer.
9200	Prints letter Y at end of last line.
9210	Stores number of characters loaded into buffer in location 158.
9260-9320	Scan displayed line to convert characters into ASCII and stores it in the buffer. These lines also detect the line end by recognising X or Y.

ANYTHING IBM CAN DO...

It's 16 BIT. It's IBM PC compatible. And it's here now!

Icarus have a new personal computer manufactured by Columbia Data Systems of the USA complete with its own software and hardware. It will also run all software and hardware produced for the IBM PC.

At first glance, this machine may bear a certain resemblance to its mentor.

That's before you switch it on.

What a performance!

Appearances can be deceptive as a look at the credentials of the Columbia PC machine will tell you.

Operating system software is MS-DOS® or CP/M 86®.

Its 16 BIT 8088 hardware configuration provides 128 K RAM with parity, two RS-232 serial ports, Centronics parallel printer port, DMA controller, eight levels of interrupt, dual floppy disc system with one megabyte storage, Winchester interface and eight expansion slots.

Not bad for the standard version.

Examine your options

Columbia like to leave your options wide open. So they've included 256K RAM boards, 8087 arithmetic co-processor for high speed math functions, dual RS-232/RS-422 boards, and a Winchester disc based system.

The Columbia PC's expansion capability can easily accommodate just about any imaginable hardware configuration including one megabyte RAM and 10 megabyte disc.

16 BIT IBM PC COMPATIBLE

It's here now

And you know the other great thing about the Columbia PC? It's here now. Available off the shelf. And Icarus have a nationwide dealer network waiting to start deliveries.

For further details, or if you would like to become a dealer yourself, contact

ICARUS

Computer Systems Ltd.

Icarus Computer Systems Ltd., Deane House, 27 Greenwood Place, London NW5 1NN.
Tel: 01-485 5574. Telex: 264209

SEE IT ON STAND NO. 401 AT THE PCW SHOW

MS-DOS is the trademark of MICROSOFT CP/M is the trademark of Digital Research

COME AND MEET VIC
AT YOUR NORTH LONDON VIC CENTRE

SALE — VIDEO GENIE — SALE

EG3003	£240.00	EG400	£195.00
GENIE I	£269.00	EG3020	£35.00
GENIE II	£269.00	EG3022	£55.00
EG3014	£185.00	EG3015	£85.00

VIC SINGLE FLOPPY DISK

* 174K Byte Storage
* Direct Interface to VIC
* Direct Compatibility with printer

ONLY £335.00

VIC CASSETTE DECK

* Direct Interface to VIC
* Tape Counter

ONLY £34.00

supplied with Free cassette with 6 Programmes

VIC EXPANSION UNIT

★ 7 Slot expansion board, for: extra memory, Programming, Games, etc.

★ Modulator Holder

★ Enclosed Power Supply

EXPANSION UNIT with LID ONLY £82.00

**** SPECIAL SUMMER OFFER ****
EXPANSION UNIT WITH LID AND MACHINE CODE MONITOR CARTRIDGE **ONLY £100.00**

VIC 20 COMPUTER

* A Typewriter Keyboard.
* 5K memory-expandable to 29.5K.
* 8 programmable function keys.
* High resolution graphics
* Music in three voices and three octaves, as well as sound effects
* Eight border colours and sixteen screen colours

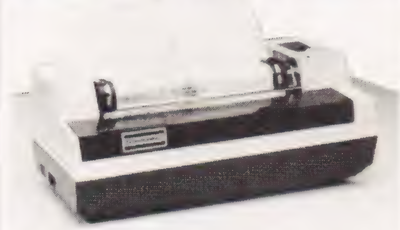
**** SPECIAL SUMMER OFFER ****
VIC 20 COMPUTER
ONLY £165.00
PLUS FREE CHOICE OF A GAMES CARTRIDGE

SALE — UK101 KITS — SALE

P.P.I.	£19.95	Colour
Prog. Sound	£19.95	£59.95
P10 Eprom	£19.95	Memory
JI Expander	£ 9.95	£79.95

All Kits now Ready Built and Tested

VIC PRINTER



* 80 characters per line
* 30 characters per sec
* Dot Matrix printer
* Tractor Feed

ONLY £185.00

CARTRIDGES

8K Memory **£34.00**

16K Memory **£59.00**

Programmers Aid **ONLY £27.50**

Super Expander High Resolution graphics with 3K RAM **ONLY £27.50**

*** SPECIAL SUMMER OFFER ***
MACHINE CODE MONITOR
ONLY £25.00

TERMS OF BUSINESS

Please add 15% VAT to all Prices. Deliver charged at Cost. Prices valid for coverdate of this magazine. Phone or Send your Order **TODAY** using

ACCESS OR BARCLAYCARD

Please send me a copy of your **'VIC LIST'** containing Software, Hardware and Books for the VIC 20

NAME ____________________

ADDRESS ____________________

____

____________________ C.T.

CHROMASONIC electronics

48 JUNCTION ROAD, ARCHWAY, LONDON N19 5RD 100 yds FROM ARCHWAY STATION TEL: 01-263 9493/01-263 9495 TELEX: 22568..

A P Stephenson

CONNECTIONS

Keeping your micro in time with itself is a matter for counters and clocks.

Quite apart from the exotic peripheral devices we've been looking at over the last couple of months, there are a number of basic logic devices which are just as essential to the designer.

The heart of any computer system is its clock — this is the signal which keeps every single piece of the hardware operating in unison. The generation of clock signals, and the ways in which they can be counted and sub-divided, will be the topic under discussion in this month's episode.

TAKING TIME

Almost as famous as the op-amp is the class of integrated circuits commonly known as 'Timers'. The most popular (because it is cheap and abundant) is the 555 Timer. Figure 23 shows the pin connections and Fig. 24 shows the connections when used as a triggered timer. A negative-going pulse on pin 2 (the trigger input) causes the output on pin 3 to rise sharply to a HIGH. It remains HIGH for a certain time 'T' and then falls sharply to its resting state again, LOW. The values of the resistor R and the capacitor C determine the time interval T in accordance with the following formula:

$$T = 1.1\ CR$$

Example
If C is 1uF and R is 1M, T is 1.1 seconds.
If C is 100pF and R is 1M, T is 110 microseconds.

When choosing values, it is not wise to have R greater than about 1M (preferably less); the lower limit for R is 1k0.

The 555 as an oscillator The term 'clock' has been mentioned several times, Fig. 25 shows how a 555 can be used to generate clock pulses at any desired frequency (within the limits set previously).

The clock signal will contain a HIGH level t_1 and a LOW level t_2. These values can be designed using the following two formulae:

$$t_1 = 0.7\ (R_A + R_B)\ C$$

$$t_2 = 0.7\ R_B\ C$$

The 555 timer is not really suitable if time periods of minutes or more are required because the values of R and C become prohibitively large. Values of C higher than about 1uF are normally of the electrolytic variety . . . not particularly noted for close tolerance! One way out, if long time delays are required, is to use two 555s in cascade ; the output of the first acting as a trigger to the second.

The better solution is to use a more sophisticated timer chip. The ZN1034E is described as a 'Precision Timer', but the word is, of course, meaningless unless the degree of precision is stated. Nevertheless, it is a useful chip, using a 12-bit counter and an oscillator which is built in. Thus, a moderately small pair of CR values sets the oscillator frequency which is used as the counter input. Because a 12-bit counter resets to zero after 4096 input pulses, the 'output' changes state at this point. The total time delay between the trigger pulse (which starts the oscillator) and the change of output state is given by the formula:

$$T = 2736\ CR \text{ seconds}$$

Example: If C = 1uF and R = 1M, T = 2736 seconds = 45 min approx.
If C = 0.01uF and R = 20k, T = 0.55 seconds.

The ZN1034E is a 14 pin chip and contains, amongst other things, a built-in voltage regulator and facilities for more precise calibration. However, for the purpose of these notes, it will be sufficient to show its use in a simple timer circuit suitable for TTL triggering and TTL driving from the output pin. Such a circuit with pin numbers is shown in Fig. 25. The remaining pins can be left unconnected.

SHIFTING AND COUNTING

There is a various assortment of shift registers and counters in the TTL catalogues . . . in fact, there are possibly too many because the task of choosing a particular specimen is made all the more difficult. A danger which is ever present is the risk of choosing a 'rarity', however, these are unlikely to be obtainable off-the-shelf. The examples treated here will be restricted to a few 'standards' which most suppliers would almost certainly stock.

Shift Registers The primary use of a shift register, ***outside*** the computer, is to convert data in parallel form to serial or ***vice versa***. Thus, it is possible to supply the shift register with eight bits of binary data in 'parallel' (all at once) and shift each bit out, one after the other, along a single wire by means of clock pulses. Conversely, data can be shifted in,

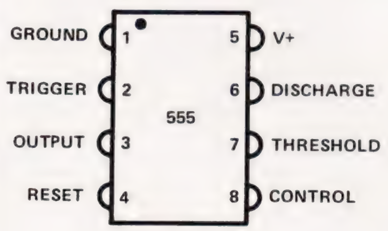


Fig. 23. The pin connections of a 555 timer.

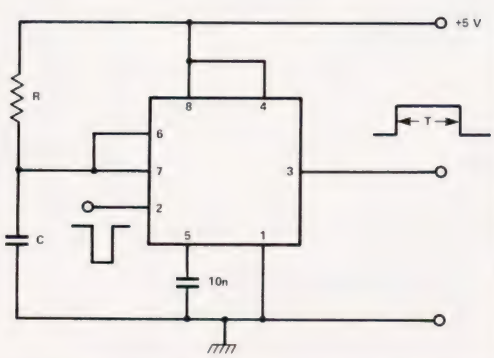


Fig. 24. The 555 wired up as a simple timer.

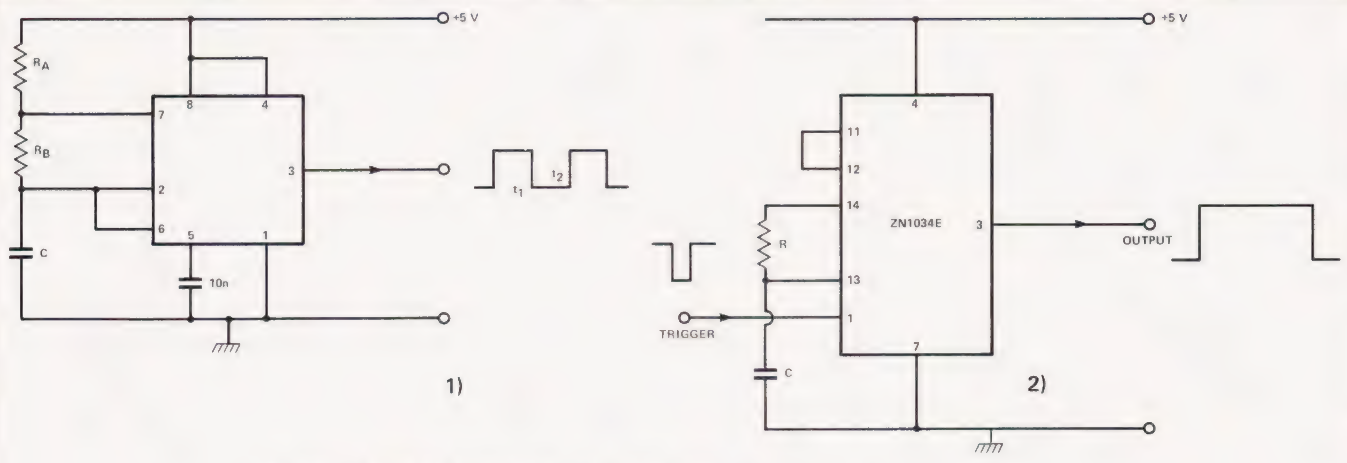


**Fig. 25. 1) The 555 operating as an oscillator.
2) The more sophisticated ZN1034E device used in timer mode.**

one bit at a time, until the eight bits have been entered and available henceforth in parallel. Some computers have such shift registers built into the input/output system but even so, the need can arise for an extra external chip.

The 74165 PISO is an eight-bit Parallel In Serial Out shift register, hence the abbreviation to PISO. The pin connections are shown in Fig. 26. To enable better understanding of the 74165, Fig. 27 gives a functional diagram.

The control line 'shift/load' is first held LOW to allow the parallel inputs to latch in. The control is then made HIGH allowing the clock pulses to shift the latched data out on the 'serial out' line. Only eight clock pulses must be sent . . . or you will be pushing out strings of 0s on the end. The clock input can be stopped at any time by setting the 'inhibit' line LOW. Notice that there is an alternative serial output line available to give the inverted data if required. There is also a provision for feeding a serial input if required, in which case the chip would act as a SISO!

The 74164 SIPO is a Serial In Parallel Out shift register (SIPO), handling eight bits; Fig. 28 shows the pin connections. This chip has a relatively simple action and a function diagram is considered unnecessary. Serial data can be clocked in on either of the two 'serial input' pins (one of them accepts an inverted input). The clock must be stopped after eight pulses, which will leave the data latched in and subsequently available on the eight parallel outputs.

Counters As with shift registers, TTL catalogues abound with various counters so again, only those which are of universal appeal will be treated. There are, however, two major divisions:

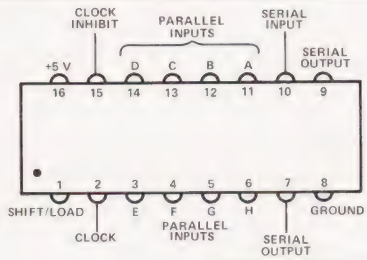
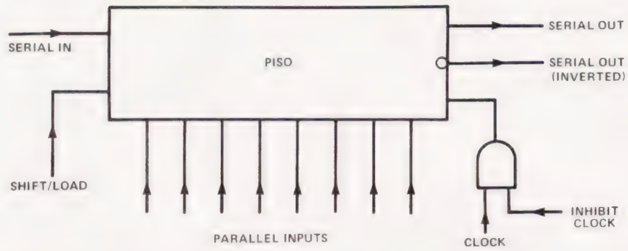


Fig. 26. The pin connections of the 74165 PISO device...

Fig. 27. ...and its corresponding function diagram.



'binary' counters and 'BCD' counters (BCD means 'Binary Coded Decimal'). BCD counters are four-bit, the output pattern changing with each clock pulse through the series 0000,0001,0010 . . . . . 1001 (0 to 9 decimal). The tenth pulse automatically resets the counter back to 0000 and the sequence repeats. Observe that the six combinations 1010 to 1111 (10 to 15) are not allowed to be present in the system. BCD counters are 'human oriented' because of the inherent decimal scheme.

So-called 'Binary' counters sequence from 0000 to 1111 and reset back to 0000 on the sixteenth pulse. Thus, all 16 combinations are used — so it may be considered as a hexadecimal counting system. Binary counters are ideal for computers (which 'think' in binary or hexadecimal) but inconvenient for humans. It would seem, therefore, that if the output of the counter is to be read by humans, BCD counters are advisable. If humans are unimportant, use the more efficient binary counters which don't waste those six combinations at the end.

Most counters are supplied with a set of 'preset' inputs which are useful if a count is to start at some pre-determined value rather than the normal 0000 start. Where this facility is available, there will be a pin marked 'Preset Enable' (or something on these lines) which allows the preset inputs into the counter. Thus, if the pattern 0011 is preset into the counter, the next clock pulse will take the count to 0100. Some counters provide a choice of 'count-up' (which is the normal sequence) and 'count-down', ie each clock pulse reduces the count by one. With down-counters, the reset state is a 'full-up' state not 0000. Thus, with four-bit binary down-counters, the reset state is 1111 but with BCD counters it is 1001 (decimal 9); Fig. 29 shows the function diagram of simple four-bit counter.

Figure 30 shows the pin connections of the 74193 which is a more sophisticated example, and can be used to count up or down depending on whether the pulses are applied to pin 5 or pin 4. Note that when used as an up-counter, the connection to the next higher ▶

order stage is called the 'carry' but called the 'borrow' when used as a down-counter. Pin 11 is normally left HIGH for normal counter action. When this is held LOW, the count is stopped and the data on the preset inputs is latched into the counter, over-writing any previous count state. When pin 11 is turned LOW again, the count proceeds from the present state.

The 74192 is identical to the 74193 except it is a BCD counter.

THE GREAT DIVIDE

It is often required to 'divide' a frequency by a certain fixed ratio. For example, if the clock frequency available is 1 MHz, a 'divide by two' stage would produce a 500 kHz signal accurately **synchronised** to the input signal.

Providing the division ratio required is a 'digital' number (2,4,8,16, 32, etc) normal binary counters can be used without modification. To appreciate this, refer to the simple counter shown in Fig. 29, and imagine each output is connected to a lamp indicator and a slow frequency is applied to the input. Lamp Q_D (the LSB) would wink slowly, in fact at half the frequency of the input. Lamp Q_C would flash at ¼ the frequency, lamp Q_B at ⅛ and lamp Q_A at 1/16 the frequency of the input. Thus, if, for example, the clock input is to be divided by four, the output would be taken from Q_C and the remaining outputs left unused. It should be mentioned here that if a divide by two stage is all that is required, it would be more economical to use a single JK flip flop.

If the division ratio required is not a normal digital number (say 10), it can be wangled using a NAND gate on the reset line to terminate the count prematurely at the instant the tenth pulse enters the counter. Figure 31 is an example showing how a divide-by-seven stage can be engineered. When the count reached seven (a'1' and a'2' and a'4'), the NAND output goes LOW and resets the count to 0000 again.

Note the reset pin is marked R with a negator bar over the top. This is common practice indicating that the reset line demands a LOW to energise it (known as an 'active LOW' function). This is why NAND was used rather than AND. It may be noted that a binary counter can be transformed into a BCD counter using a circuit similar to this. The only change required would be a two-input NAND fed from the '8' and the '2' lines.

It would be ungrateful to end this section on counters without mentioning the most common (notorious ?) counter chip of them all . . . the 7490. This chip has been out a long while (some say it was first seen at Waterloo) and has been a faithful friend to all those who dabble in logic. It can be used as a normal BCD counter or a divide-by-two or four or five or eight or ten without additional gating. Figure 32 shows the pin connections and the internal logic which will illustrate its versatility.

THE FINAL INSTALMENT

Next month we'll be taking a look at the least complex of all the logic devices, the simple gates. Despite all the built-in circuitry the various complex devices possess, it is still essential to be able to perform simple logical functions on the signals which they produce. In the next part of this series we'll be going right back to the basement of digital devices — the plce you *might* have expected us to start!

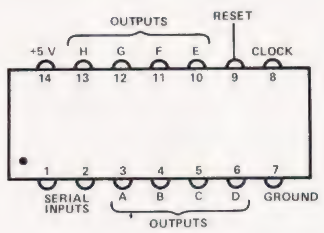


Fig. 28. The pin connections for the

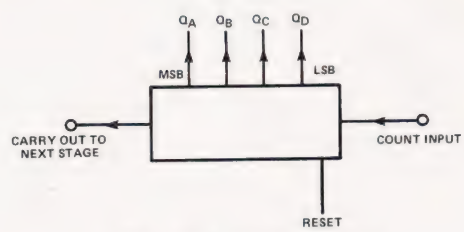


Fig. 29. A simple four-bit counter.

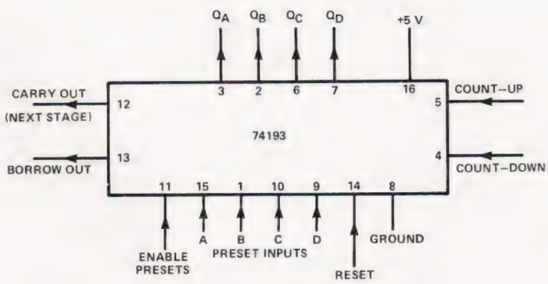


Fig. 30. The 74193 up/down counter features preset inputs.

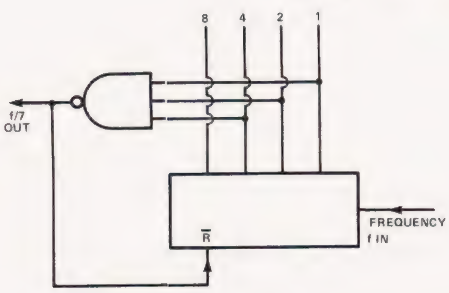


Fig. 31. By applying some simple logic to a counter's outputs we can achieve a divide by seven system.

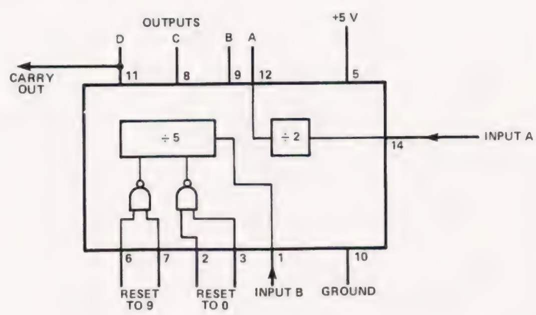


Fig. 32. Possibly the most famous of all counters, the 7490 device.

The new Dragon 32. So well designed, you'll even understand this ad.

If you're already a computer expert, may we refer you to the box of technical specifications displayed opposite.

If you're not, may we refer you to the new Dragon 32 Family Computer. A computer so easy to understand, you won't understand why all the others seem so difficult.

And the new Dragon 32 costs under £200.

32K RAM FOR UNDER £200?*

When you're comparing computers, the first thing you need to know is the size of the memory. In plain English, the Dragon has approximately 32 thousand units of Random Access Memory. (32K RAM for those who prefer to be blinded by science.) This means that the Dragon's memory is at least twice as powerful as its competitors'.

With a memory this powerful, the amount of information the Dragon can store is literally vast. But the Dragon doesn't just make it easy to store information. It makes it easy to use, too.

USER-FRIENDLY?

You may have heard of the term 'user-friendly.' Reverting to plain English once more, this means simply that the computer will go out of its way to understand you, rather than vice-versa.

The Dragon 32 is so user-friendly, it practically licks your hand.

You tap (literally) its vast resources through a beautifully-designed keyboard that's as easy to use as a typewriter.

On this keyboard, you type in a language which is surprisingly close to the English you talk every day. The Dragon 32 will receive your order. Understand it. Send it to the appropriate section of its massive brain. And then display the appropriate information on your screen. All before you can say 'gobbledy-gook'.

*TV not included in price.

SPECIFICATIONS
6809E MICROPROCESSOR. Pet, Apple, Atari 400, BBC Micro, and VIC 20 still have the less powerful 6502.
32K RAM (as standard). At least twice the power of similarly priced machines. Expandable to 64K RAM.
EXTENDED MICROSOFT COLOUR BASIC (as standard). Featuring: ADVANCED GRAPHICS (set, line, circle, paint, print, draw, rotate and print using). ADVANCED SOUND 5 octaves, 255 tones. AUTOMATIC CASSETTE RECORDER CONTROL. FULL EDITING with INSERT and DELETE.
9 COLOUR, 5 RESOLUTION DISPLAY.
USE WITH ANY U.H.F. TV and/or separate P.A.L. monitor.
PROFESSIONAL QUALITY KEYBOARD. Typewriter feel. Guaranteed for 20 million depressions.
PRINTER (Centronics parallel).
JOYSTICK CONTROL PORTS.

FEATURE \ PRODUCT	DRAGON 32	SINCLAIR SPECTRUM	ACORN ATOM	VIC 20	TI 99/4A	BBC MICRO 'A'
PRICE	£199	£125	£175	£190	£199	£300
STANDARD RAM SIZE	32K	16K	8K	5K	16K	16K
STANDARD AVAILABLE RAM FOR HIGH RESOLUTION GRAPHICS	26K	9K	N/A	N/A	14K	3K
EXTENDED MICROSOFT BASIC AS STANDARD	YES	NO	NO	NO	NO	NO
PROFESSIONAL-TYPE KEYBOARD	YES	NO	YES	YES	YES	YES

FIRE YOUR IMAGINATION.

Learning how to use the Dragon 32 won't cause you to experience any problems. Learning what you can use it for will cause you to experience something entirely different.

Delight. Surprise. Fascination. And challenge.

The Dragon offers a range of some of the most popular computer games in the world. From those celebrated space battles to mind-boggling adventures in seemingly unfathomable dungeons and caves.

As if by magic, a simple typed message will command the Dragon to create your own drawings. Then it will colour and paint them in 9 colours.

And it's clever enough to create virtually any image you want - circles and arcs as well as straight lines.

The Dragon will also play and compose music with you, with a range of 5 octaves. And it works with any UHF TV or PAL monitor.

LEARNING THROUGH PLAYING.

All of this makes the Dragon the ideal machine to build your children's interest in the world of computers as they become increasingly more vital. School-children already enjoy using computers.

The Dragon is the first computer specifically for the family - so by enjoying yourselves at home, you and your children can soon become expert enough to create your own programs.

BRILLIANTLY SIMPLE GUIDE.

The Dragon is living proof that you don't have to be an expert in computerspeak to be an expert in computers. It comes with the easiest-to-understand instruction manual ever written for a home computer.

Every step, every explanation, is made clear - even if you're a beginner. In minutes, it will show you how to write a simple program. Within hours, you'll be fascinated. And from then on, you'll continue to be astounded by the new world which the Dragon's power and versatility will open up to you.

See the new Dragon 32 in your High Street. At under £200, it's not just the first family computer. It also has all the features an expert could wish for.

Except perhaps the jargon.

DRAGON 32
The first family computer.

To: Jean Webster, Dragon Data Ltd, Queensway, Swansea Industrial Estate, Swansea, Glamorgan SA5 4EH.
Tel: 0792 580651.

Please send me further information about the Dragon 32.

Name ____________________

Address ____________________

____

____________________ CT

A member of the Mettoy Group of Companies.

D S Peckett

DATA STORAGE

The final part of our series on efficient storage.

This is the third, and final, part of our short series on ways of making better use of cassettes as a data storage medium for microcomputers. We have seen that, in some micros at least, tape-based data storage is very inefficient and so have taken a look at how we might get more into a given length of tape than the computer's designers intended. The advantages of this are that read and write times can be considerably reduced and the risk of errors is also lessened.

So far in the series, I have explained just how data is stored and provided some routines for packing strings and integers onto tape more densely. This month, I will give you some ideas about storing floating point (FP) variables and go on to explain how, in some circumstances, we can make considerable improvements on the general-purpose routines I have provided so far.

As in the earlier articles, I'm including complete subroutines to carry out the various tasks. The routines are written so that they can be incorporated, 'cook-book' fashion, into your own programs; all you have to do is avoid the use of variable names beginning with Z. Each routine is self-contained and will not interfere with any other in the series. To pass information between the subroutines and the calling program, I use two global variable names: the array DT() which contains, or will receive, the data and N, which contains the number of data items to be handled.

FLOATING POINT VARIABLE STORAGE

Last month we saw that a typical micro requires anything up to 12 bytes when it stores a floating point number on tape by converting it into string format. However, since each FP number is stored in only four bytes of RAM (in the case of Level II BASIC), there must be a way of improving the situation.

We had a similar problem with integer storage and overcame it by using BASIC's AND function to pull groups of bits out from the number. We could then treat these bit groups as pseudo-characters which could be written to tape. Unfortunately, FP numbers are not quite so easy to handle. In particular, it is not practical to directly mask out groups of bits in order to form characters.

Some computers, however, do provide a way of doing a similar job. The TRS-80 and Video Genie are good examples, since they possess the VARPTR() command. We can use this word to find out where in RAM a number is stored; in the case of FP and integer numbers it is easy, PT = VARPTR(X) gives the address of the first (ie lowest address) byte of the group which contains the value of X. Thus, if X is an FP number (Fig. 1), PT is the address of byte 1, and the remaining three bytes are stored at (PT + 1), (PT + 2) and (PT + 3). Things are not quite so easy with strings, but that is irrelevant at this point.

Once we know where in memory a number is stored, we can use PEEK() to read any or all of the bytes which make it up. From that point, it is only a short step to realise that cunning use of PEEKs and ANDs will allow us to read any given bit(s) in an FP number. Once we can get at the bits, we can combine them into pseudo-characters before writing them to tape.

How many characters do we need to hold an FP number? At first glance the answer looks like four, but think again... Each byte of the four containing X can have any value from 0 to 255. Certain of those values, when treated as characters, will look to the computer exactly the same as I/O control characters such as ',' and ':'. Writing or reading them to or from tape would be a sure way of crashing the system.

We had the same problem when we handled integers, and overcame it by setting the most significant bit (MSB) of each pseudo-character to 1. This, of course, allowed us to store only seven genuine data bits in each character. A quick bit of mental arithmetic tells us that we need five bytes to code up each FP variable. Not quite as good as we might like, but definitely an improvement on the maximum of 12 which the computer would need if we left it to its own devices.

PACK IT

That should be enough groundwork before we move on to Listing 1, which packs FP numbers onto tape. Lines 10000-10040 are basically housekeeping to set up the routine; Z1 is set to 49 at the start since we can only save a maximum of 249 bytes in any Level II tape block (INT(249/5) is 49).

Line 10050 reads the variable to be saved into a dummy variable, ZN, and line 10060 finds where ZN

Fig. 1. A floating point number showing where the exponent and mantissa are stored.

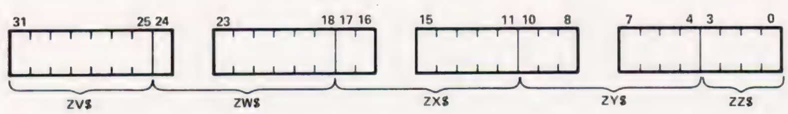


Fig. 2. The way the 32 bits of ZN are distributed among ZV$-ZZ$.

Fig. 3. The first 16 bits of ZN.

is stored in memory. Lines 10070-10110 then form the five pseudo-characters holding the number. Each of the lines uses the same sort of bit-masking and shifting technique I described last month to pull out the bits it needs and to get them into the correct places. To refresh your memory, we use AND to pull out the bits we need, PEEKing into memory to find the bytes which contain them. In order to pack the bits properly into the pseudo-characters, we may need to shift them left or right; multiplying by 2^n shifts them n places left while division by 2^n moves them n bits right. The + 128 simply sets the MSB of each character. Figure 2 shows how the 32 bits of ZN are split among ZV$-ZZ$; ZV$-ZY$ each contain seven bits of the original while ZZ$ holds the remaining four bits.

Having formed the five characters, they are added, in the correct order, to the string (ZA$) which will eventually be output to tape. Having done that, the program goes back to handle the next variable, and so on.

Of course, it's no use storing the numbers on tape if we cannot get them back again, and Listing 2 is my offering to do the job. The top and tail of the routine are, once again, just there to act as the bread which holds the real meat in the middle. The first important job is done by line 11060, which uses the MID$() function to isolate the group of five bytes which contains the coded variable we are interested in. Once we have these bytes, it is simple for lines 11070-11110 to split them back into the five individual characters (ZV$-ZZ$) which we originally used when we wrote the value to tape.

Having extracted the data, the fun really starts. First of all, we create a dummy variable (ZN) to receive the data. We can then go through what is, essentially, the reverse of the write process in order to recover the original (saved on tape) value. We use AND to pull the data bits out of ZV$-ZZ$, shift them to the correct positions by multiplication or division, and then POKE the result into the correct byte of ZN.

Once that bit of jiggery-pokery is complete, it is simplicity itself to assign the new value of ZN to the correct element in the data array DT() (line 11180).

Just in case my deviousness has left you slightly bemused, let me try to make the process a bit easier to follow. Suppose that we are writing data to tape and the first 16 (Fig. 3) of ZN are:

```
abcd  efgh  jklm  npqr
```

The values of the last 16 don't really matter here. Then, going through the data-extraction process in exactly the same way as the subroutine:

```
PEEK(ZP)=abcd efgh
PEEK(ZP) AND 254=abcd efg0
(PEEK(ZP) AND 254)/2=0abc defg
(PEEK(ZP) AND 254)/2+128=1abc defg
ZV$=CHR$(1abc defg)

PEEK(ZP) AND 1=0000 000h
(PEEK(ZP) AND 1)*64=0h00 0000
PEEK(ZP+1) AND 252=jklm np00
(PEEK(ZP+1) AND 252)/4=00jk lmnp
ZW$=CHR$((0h00 0000)+(00jk lmnp)+128)
ZW$=CHR$(1hjk lmnp)
```

When we come to get the data back, we reverse the process:

```
ASC(ZV$)=1abc defg
ASC(ZV$) AND 127=0abc defg
(ASC(ZV$) AND 127)*2=abcd efg0
ASC(ZW$)=1hjk lmnp
ASC(ZW$) AND 64=0h00 0000
(ASC(ZW$) AND 64)/64=0000 000h
1st byte of ZN=(abcd efg0)+(0000 000h)
1st byte of ZN=abcd efgh
```

and we are right back where we started. Nothing to it, is there?!

Procedures like those in Listings 1 and 2 can show substantial improvements in read and write times if you compare them with the computer's built-in routines. Last month I gave sample programs which coded FP numbers as simple strings in order to save them; they could write 500 random FP numbers to tape in 220 seconds and read them back in 236 seconds. By way of comparison, Listing 1 can write 500 numbers in 180 seconds and Listing 2 can recover them in 184 seconds. This represents savings of 18.2% and 22.0% respectively, or a shortening of the whole read/write cycle by 20.2%.

FURTHER IMPROVEMENTS

All of the routines which I have described in the series so far have been general-purpose ones; that is to say, they can deal with any integer or FP number (as appropriate) they may meet. Very often, however, we know from the start that the numbers we must save will fall into a limited, and precisely-defined range. For example, it may well be necessary to save a large set of marks from some form of exam — it is a fairly safe bet that every one of those marks falls in the range 0-100 and is an integer.

At times like this, it is a waste of effort (and tape) to use a routine designed to handle integers Between − 32768 and + 32767 to store the marks — a much better way is at hand. Don't forget that, fundamentally, we are using groups of bytes to store numbers in a coded format. It is just for convenience, and to overcome a limitation of Level II BASIC, that we assemble these bytes into long strings (ZA$ in the routines) before writing them to tape — it might very well be that other types of micro will not need the strings.

No, the fundamental storage block is the byte. In each byte, the MSB must be set to 1 in order to overcome I/O procedure problems, but that leaves seven bits available to save the data held in each byte. Each byte can thus hold any one of 128 different values, such as 0-127, 2000-2127, − 50 to + 77, etc, and there is the clue to a much better way of saving our marks.

The value of each mark can be held in a single byte, with the advantage that a total of 249 separate marks can be stored in each Level II tape block. (Compare that to the maximum of 41 which would be possible if we stayed with crude number-to-string conversions.) Listing 3 is an example of a subroutine which will save numbers between 0 and 127 on tape.

It follows the basic pattern we have established in the series with the opening and closing lines being concerned with the housekeeping of the routine, such as controlling its loops. The routine is, in fact, closely related to Listing 1, as you will see if you compare the two. Line 12000 defines Z1, the items per block counter, as 249, the number of characters which can be saved in a single Level II tape block. This number would, of course, probably be different for other micros. Line 12050 is the heart of the routine, converting a single variable to a single pseudo-character with its MSB set. Each character is added to the write block, ZA$, as it is created, and ZA$ is written to tape each time it fills. It is as simple as that!

Listing 4 is the read counterpart of Listing 3 and, once again, follows the now-standard format. Its heart is lines 13060 and 13070, which pick out each character, convert it to an integer, strip off the MSB and save it to the correct place in DT(). Remember that the routine is written to be easy to follow and a practical version would certainly use much more compact code.

Both of these routines are for integers in the fundamental range of 0-127 which a single byte allows, but it is easy to use suitable scaling constants to adapt them to handle other ranges. For instance, ▶

suppose we had to store numbers in the range −50 to +50. Simply change line 12050 to:

```
12050  ZZ$=CHR$(DT(Z4)+178):REM ** Add
       "+50" offset
```

and line 13070 to:

```
13070  DT(Z3*Z1+Z5)=ASC(ZZ$)-178:
       REM ** Subtract the offset
```

and the job is done. Naturally, the offset can change magnitude and sign to suit the precise range which has to be saved.

To give an idea of the savings which are possible from this approach, last month's crude, string-based, integer handling routines could write 500 random integers in 118 seconds and recover them in 132 seconds. The corresponding times for Listings 3 and 4 are 37 seconds and 41 seconds respectively — reductions of 68.6% and 68.9%!

We can go even further with this technique. By using two bytes, each with seven bits available for data, we can store (2^{14}), or 16384, different integers; for example, the basic range is 0-16383. We can even stay with the routines of Listings 3 and 4, if we make some simple changes to them.

The first step is to adjust Z1 in lines 12000 and 13000 to be 124 (INT(249/2)), since we cannot store so many numbers in each block. Having done that, generate the write routine by means of the patch in Listing 5. It uses integer division to separate the number to be saved into its high and low halves — if you prefer, these are, respectively, the quotient and remainder of the sum (DT(Z4)/128). These are then converted, as normal, into the characters ZY$ and ZZ$ which are in turn saved in the usual way.

To get back the data, use the patch of Listing 6 to modify Listing 4. By now it should be easy to follow what is happening. It pulls out the high and low characters, converts them to numbers, strips off the MSBs and re-forms the original answer which it puts back into DT().

Performance? The patched routines will save 500 random integers in 74 seconds and recover them in 73 seconds. These represent savings of 37.3% and 44.7% respectively over last month's crude approach.

The double-byte routines can, naturally, be modified to handle any range of integers as long as the difference between the most-positive and the most-negative values is not greater than 16383.

CONCLUSION

Now, you have a set of routines which can save string, integer and FP data on tape in a much more efficient way than is possible via the standard routines available from most micros. I have tried to cover all the most important possibilities, and to give you some hints as to how the approach could be extended to handle other related problems.

The routines are designed to be used in your own programs and I have made them as transportable and self-contained as possible. They will, inevitably, need some modification to suit micros other than the TRS-80 or Video Genie but, in most cases, it should not be too difficult. ■

```
 9999  REM ** SAVE FLOATING POINT NUMBERS IN THEIR BINARY
       FORMAT
10000  Z1=49:REM ** Number of numbers we can save in one
       block
10010  FOR Z2=1 TO N STEP Z1
10020  ZA$="":REM ** Clear dummy string
10029  REM ** Make sure there are enough items to fill the
       string
10030  IF (N-Z2)>Z1 THEN Z3=Z1 ELSE Z3=(N-Z2+1)
10040  FOR Z4=Z2 TO (Z2+Z3-1):REM ** Loop Z3 times
10050  ZN=DT(Z4):REM ** Dummy variable
10060  ZP=VARPTR(ZN):REM ** Get address of dummy variable
10070  ZV$=CHR$((PEEK(ZP) AND 254)/2+128):REM ** Bits
       25-31
10080  ZW$=CHR$((PEEK(ZP) AND 1)*64+(PEEK(ZP+1) AND 252)/4
       +128):REM ** Bits 18-24
10090  ZX$=CHR$((PEEK(ZP+1) AND 3)*32+(PEEK(ZP+2) AND 248)
       /8+128):REM ** Bits 11-17
10100  ZY$=CHR$((PEEK(ZP+2) AND 7)*16+(PEEK(ZP+3) AND 240)
       /16+128):REM ** Bits 4-10
10110  ZZ$=CHR$((PEEK(ZP+3) AND 15)+128):REM ** Bits 0-3
10120  ZA$=ZA$+ZV$+ZW$+ZX$+ZY$+ZZ$:REM ** Add to dummy
       string
10130  NEXT Z4
10140  PRINT#-1,ZA$:REM ** Save the coded numbers
10150  NEXT Z2:REM ** Back for more
10160  RETURN
```

Listing 1

```
10999  REM ** UNPACK CODED FLOATING POINT NUMBERS
11000  Z1=49:REM ** Number of variables in each tape block
11010  Z2=INT(N/Z1+0.9999):REM ** Number of tape blocks to
       be read
11020  FOR Z3=0 TO Z2=1:REM ** Loop Z2 times
11030  INPUT#-1,ZA$:REM ** Read a block
11040  Z4=LEN(ZA$)/5:REM ** Number of items in this block
11050  FOR Z5=1 TO Z4
11060  ZD$=MID$(ZA$,(Z5-1)*5+1,5):REM ** Get a coded
       number
11070  ZV$=LEFT$(ZD$,1)
11080  ZW$=MID$(ZD$,2,1)
11090  ZX$=MID$(ZD$,3,1)
11100  ZY$=MID$(ZD$,4,1)
11110  ZZ$=RIGHT$(ZD$,1):REM ** Get the 5 'characters'
11120  ZN=0:REM ** Dummy variable to receive data
11130  ZP=VARPTR(ZN):REM ** Point to dummy variable
11139  REM ** Separate the bits and POKE them into place
11140  POKE ZP,((ASC(ZV$) AND 127)*2+(ASC(ZW$) AND
       64)/64):REM ** 1st byte
11150  POKE (ZP+1),((ASC(ZW$) AND 63)*4+(ASC(ZX$) AND
       96)/32):REM ** 2nd byte
11160  POKE (ZP+2),((ASC(ZX$) AND 31)*8+(ASC(ZY$) AND
       112)/16):REM ** 3rd byte
11170  POKE (ZP+3),((ASC(ZY$) AND 15)*16+(ASC(ZZ$) AND
       15)):REM ** Last byte
11180  DT(Z3*Z1+Z5)=ZN:REM ** Set the value in DT()
11190  NEXT Z5
11200  NEXT Z3
11210  RETURN
```

Listing 2

```
11999  REM ** SAVE POSITIVE INTEGERS OF VALUE 127 AND LESS
12000  Z1=249:REM ** One number per byte in tape block
12010  FOR Z2=1 TO N STEP Z1
12020  ZA$="":REM ** Initialise
12030  IF (N-Z2)>Z1 THEN Z3=Z1 ELSE Z3=(N-Z2+1)
12040  FOR Z4=Z2 TO (Z2+Z3-1)
12050  ZZ$=CHR$(DT(Z4)+128):REM ** Convert number to
       character
12060  ZA$=ZA$+ZZ$:REM ** Add to dummy string
12070  NEXT Z4
12080  PRINT#-1,ZA$:REM ** Save the pseudo characters
12090  NEXT Z2
12100  RETURN
```

Listing 3

```
12999  REM ** RECOVER POSITIVE INTEGERS UP TO 127
13000  Z1=249:REM ** Max number per tape block
13010  Z2=INT(N/Z1+0.9999):REM ** How many blocks?
13020  FOR Z3=0 TO Z2-1
13030  INPUT#-1,ZA$:REM ** Read a block of pseudo
       characters
13040  Z4=LEN(ZA$):REM ** How many numbers in this one?
13050  FOR Z5=1 TO Z4
13060  ZZ$=MID$(ZA$,Z5,1):REM ** Get one number from block
13070  DT(Z3*Z1+Z5)=ASC(ZZ$)-128:REM ** Character to
       number
13080  NEXT Z5
13090  NEXT Z3:REM ** Read another block
13100  RETURN
```

Listing 4

```
12041  REM ** MODIFICATION TO HANDLE INTEGERS UP TO 16383
12045  ZQ=INT(DT(Z4)/128):REM ** Get top half of number
12050  ZY$=CHR$(ZQ+128):REM ** Convert it to a 'character'
12055  ZZ$=CHR$(DT(Z4)-ZQ*128+128):REM ** Same for lower
       half
12060  ZA$=ZA$+ZY$+ZZ$:REM ** Add it to output string
```

Listing 5

```
13051  REM ** MODIFICATION TO INPUT ROUTINE
13050  FOR Z5=1 TO (Z4/2-1):REM ** Don't loop so many
       times
13055  ZY$=MID$(ZA$,(2*Z5-1),1):REM ** "High" character
13060  ZZ$=MID$(ZA$,2*Z4,1):REM ** "Low" character
13065  ZQ=(ASC(ZY$)-128)*128+ASC(ZZ$)-128:REM ** Number
       recovered
13070  DT(Z3*Z1+Z5)=ZQ:REM ** Save it
```

Listing 6

Subscriptions

Personally, we think you'll like our approach to microcomputing. Each month, we invite our readers to join us in an abundance of feature articles, projects, general topics, software listings, news and reviews — all to help committed micro users make more of their microcomputers.

However, if you've ever missed a copy of Computing Today on the newstands, you'll not need us to tell you how valuable a subscription can be. Subscribe to CT and for a whole year you can sit back, assured that each issue, lovingly wrapped, will find its way through your letter box.

And it's not difficult! All you have to do is fill in the form below, cut it out and send it (or a photocopy) with your cheque or Postal Order (made payable to ASP Ltd) to:

COMPUTING TODAY Subscriptions,
513 London Road,
Thornton Heath,
Surrey CR4 6AR.

Alternatively, you can pay by Access or Barclaycard in which case, simply fill in your card number, sign the form and send it off. Please don't send in your card.

Looking for a magazine with a professional approach with material written by micro users for micro users? Why not do yourself a favour and make 1982 the year you subscribe to Computing Today and we'll give you a truly personal approach to microcomputing.

SUBSCRIPTION ORDER FORM

Cut out and SEND TO :
COMPUTING TODAY Subscriptions
513 LONDON ROAD,
THORNTON HEATH,
SURREY CR4 6AR.

Please commence my subscription to Computing Today with the .... issue.

SUBSCRIPTION RATES (tick ☐ as appropriate)		
	£12.10 for 12 issues UK	☐
	£15.75 for 12 issues Overseas Surface	☐
	£35.35 for 12 issues Overseas Air Mail	☐

I am enclosing my (delete as necessary) cheque/ Postal Order/ International Money Order for £..........
(made payable to ASP Ltd)
or
Debit my Access/ Barclaycard*
(*delete as necessary)

Please use BLOCK CAPITALS and include postcodes.

NAME (Mr/ Mrs/ Miss) ....
delete accordingly

ADDRESS ....

....

.... POSTCODE ....

Signature ....

Date ....

82/8

A P Stephenson

SUBROUTINE LIBRARY

A 'starter kit' of general purpose routines to make programming easier.

As far as BASIC is concerned, a subroutine is something you 'call up' with the keyword GOSUB followed by the line number at which the subroutine starts. On completion of the allotted task, the keyword RETURN mysteriously directs control back to the calling program. In some respects, a subroutine behaves as a 'subcontractor' to the main program. And in much the same way that subcontractors will often be obliged to subcontract yet again, a subroutine may often call up another subroutine in order to complete a task.

This technique is called 'nesting' and may usually be extended to any level, subject to the limitations imposed by the area of memory known as the 'stack'. When GOSUB is used, the address of the next statement (which will be in an internal register called the Program Counter) is stored in the stack in order for the subroutine RETURN statement to know where to return to! If a subroutine calls on another, the appropriate address must again be stored.

If the level of nesting is increased too far, the poor old stack overflows and loses the last return address. The net result is either a system crash or at least an error message pointing out the memory deficiency. If an 'OUT OF MEMORY' error appears, it is not always due to actual shortage of memory — there may be several K left. The shortage is restricted to the relatively small allocation of stack memory. For efficiency purposes, the stack is normally located in Page zero or Page one of memory and is seldom more than a hundred or so bytes capacity in the majority of microcomputers. When this trouble arises, the only cure is to re-write some of the program to reduce the nesting level.

Originally, the term subroutine simply meant a block of coding intended to be used several times in the main program. If this block is written into the program every time it is needed, it is called an 'open loop' subroutine. If, on the other hand, it is written once only and used with GOSUB everytime, it is called a 'closed loop' subroutine. Although the normal closed-loop method is universal and superficially more efficient in coding time and memory usage, a time penalty is involved. If execution time is paramount, an open loop block spliced in wherever it is needed will at least make a marginal improvement and may justify the extra bout of keyboard bashing. It should be pointed out that if speed is of the essence, the subroutine should be written in machine or assembly code anyway. However, this article is concerned with BASIC subroutines of the conventional closed loop type.

WHEN I'M CALLING YOU

Some subroutines can be called up straight away with GOSUB. Others may require a few preliminary assignment statements known as the 'calling sequence'. For example, a subroutine devoted to the simple task of drawing a screen line requires no calling sequence prior to GOSUB. On the other hand, a subroutine dedicated to perform some mathematical function on X and Y will require that the variables X and Y contain the numbers to be processed. If they currently reside in say, B3 and G2, then we must re-assign as follows before using GOSUB:

```
X=B3:Y=G2
GOSUB xxxx
```

These pre-assignments constitute the calling sequence.

On return from the subroutine, it may be necessary to apply the reverse procedure:

```
B3=X:G2=Y
```

The calling sequence in this example is a case of 'parameter-passing'. It may be argued with some justification that changing variable names should be unnecessary. Why couldn't the subroutine be written with variable names to suit the demands of the calling program? This is not always possible. For example, the same subroutine may be called up to perform the process at different times on different variables. The subroutine may have been written within a collection of general purpose subroutines in which case there could have been no prior knowledge of the variable names employed in subsequent programs.

WHAT KIND?

This question is similar to the 'length of the piece of string' caper. It is sufficient to remark that anything purporting to be ***general purpose*** cannot be ***specific*** in aim. The best attitude to adopt in choosing the list is to consider the task as an upgrading of the operating system and the BASIC interpreter. A high level language itself (such as BASIC) is in reality nothing more than a complex conglomerate of subroutines. These are, of course, written in machine code but nevertheless, they are still subroutines. Many of the facilities offered are taken for granted and we only take notice of the defects or omissions. Anyone who can write a high level interpreter or compiler would certainly have my respect.

In spite of this, for reasons of ROM space, any high level language cannot cover more than a sprinkling of keywords. A set of general purpose subroutines on tape or disc is equivalent to a language upgrading. Although no new keywords are added, the effect is the same. For example, most BASICs include SIN, COS, TAN and ATN but few offer the other inverse trig functions ASN (arc sine) and ACS (arc cosine). However, if GOSUB 5000 changes X into arc sine X then the BASIC is upgraded providing, of course, you have written such a subroutine and it is resident in memory. Naturally if your inclinations are such that arc sine X would only be used sparingly, if at all, then its inclusion would be foolish. General purpose subroutines are to some extent subjective to the individual.

Apart from augmenting the BASIC vocabulary, subroutines can be used to remedy defects in existing keywords. The INPUT statement in the PET has a truly diabolical feature. If the Return key is pressed (by accident) before the requested data has been keyed in, the program breaks out into COMMAND mode. A subroutine which replaces INPUT can easily be installed to remedy this effect.

Some subroutines can be very

complex and others so simple as to be open to a charge of triviality. A simple subroutine, however, can still be a time and memory saver. For example, nothing could be more simple than a row of '-' across the screen in order to produce a line — but if many lines had to be drawn in the program, it would certainly justify a subroutine. On the other hand, it may be that a subroutine which requires an elaborate calling sequence each time might very well be counter productive. In such a case, it would be easier and more efficient to splice in the code each time it was used — an 'open loop' solution.

CHOICE OF VARIABLE NAMES...

It is good practice to choose variable names which suggest the data item (as far as it is possible with the two-character limit imposed by BASIC). When writing any form of general purpose subroutine it may be politic to ignore the rules in order to avoid clashes between the names. Unless care is taken, the subroutine may use working variables which are already used in the calling program with disastrous results. To act as a safeguard, a good plan is to use 'unusual' names in the subroutines in the belief that it would be too much of a coincidence to choose such names in the calling program. Using a high digit as the second character is also a good idea (thus N8 or K8).

...AND LINE NUMBERS

If subroutines are placed at the head of the program with low line numbers there is, in most BASICs, a speed advantage because the GOSUB search commences from the top downwards. However, it is 'tidier' to place them down the bottom end. Where subroutines of the general purpose variety are concerned there is another reason why they should be at the bottom

Program Listing

```
50000 REM ** ******************
50010 REM ** PRINT N8 TO P8 DECIMAL PLACES
50020 REM ** WITH DECIMAL POINT POSITIONED
50030 REM ** AT TAB T8
50040 P8=ABS(P8):T8=ABS(T8)
50050 N8=INT(10^P8*N8+0.5)/INT((10^P8))
50060 N8$=STR$(INT(N8))
50070 IF INT(ABS(N8))<1 THEN N8$="1"
50080 PRINT TAB(T8-LEN(N8$));N8
50090 RETURN
50100 REM ** ******************
50110 REM ** PRINT BORDER
50120 A8=32768:B8=40
50130 FOR C8=0 TO 39
50140 POKE A8+C8,42:POKE A8+C8+B8*20,42:NEXT
50150 FOR C8=1 TO 20
50160 POKE A8+B8*C8,42:POKE A8+B8*C8+39,42:NEXT
50170 RETURN
50180 REM ** ******************
50190 REM ** FLASH "INVALID DATA"
50200 PRINT CHR$(147):FOR T9=1 TO 5
50210 PRINT TAB(12)"**************"
50220 PRINT TAB(12)"*INVALID DATA*"
50230 PRINT TAB(12)"**************"
50240 FOR T8=1 TO 200:NEXT:PRINT CHR$(147)
50250 FOR T8=1 TO 200:NEXT
50260 NEXT
50270 RETURN
50280 REM ** ******************
50290 REM ** VERTICAL BAR OF LENGTH N8
50300 REM ** AT CO-ORDINATE X8
50310 IF N8>25 THEN N8=25
50320 A8=33728+X8
50330 FOR Z8=0 TO N8-1
50340 POKE (A8-40*Z8),117:NEXT
50350 RETURN
50360 REM ** ******************
50370 REM ** CONVERT N8 TO BINARY IN B8$
50380 T8=2:B8$="":R8$="0":N8=ABS(N8)
50390 IF N8=0 THEN 50450
50400 D8=N8/T8:I8=INT(D8)
50410 IF D8<>8 THEN R8$="1"
50420 IF D8=I8 THEN R8$="0"
50430 B8$=R8$+B8$:N8=I8
50440 GOTO 50390
50450 RETURN
50460 REM ** ******************
50470 REM ** CONVERT N8 TO HEX IN B8$
50480 REM ** REQUIRES DIM H8$(16)
50490 RESTORE:N8=ABS(N8)
50500 FOR E8=0 TO 15:READ H8$(E8):NEXT
50510 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
50520 B8=16:B8$=""
50530 IF INT(N8)=0 THEN 50580
50540 N8=N8/B8
50550 R8=INT(0.00001+B8*(N8-INT(N8)))
50560 A8$=H8$(R8):B8$=A8$+B8$:GOTO 50530
50570 GOTO 50530
50580 RETURN
50590 REM ** ******************
50600 REM ** CONVERT FOUR DIGIT HEX IN A8$
50610 REM ** TO DECIMAL IN D8
50620 D8=0:H8$="123456789ABCDEF"
50630 L8=LEN(A8$)
50640 FOR A8=4 TO 1 STEP-1
50650 F8$(A8)=MID$(A8$,A8,1)
50660 FOR N8=1 TO 16
50670 IF F8$(A)="0" THEN 50700
50680 D8$=MID$(H8$,N8,1)
50690 IF F8$(A8)=D8$ THEN D8=D8+N8*(16^(L8-A8))
50700 NEXT
50710 NEXT
50720 RETURN
50730 REM ** ******************
50740 REM ** MERGE SORT E8 ITEMS IN
50750 REM ** ARRAY A8$(N8)
50760 G8=INT((LOG(E8)/LOG(2))+1):F8=2^G8
50770 FOR H8=1 TO G8:F8=F8/2
50780 C8=0
50790 FOR N8=1 TO E8-F8
50800 IF A8$(N8)<=A8$(N8+F8) THEN 50830
50810 B8$=A8$(N8):A8$(N8)=A8$(N8+F8):A8$(N8+F8)=B8$
50820 C8=1
50830 NEXT
50840 IF C8=1 THEN 50780
50850 NEXT
50860 RETURN
50870 REM ** ******************
50880 REM ** COMPUTE SINH N8
50890 N8=(EXP(N8)-EXP(-N8))/2
50900 RETURN
50910 REM ** ******************
50920 REM ** COMPUTE COSH N8
50930 N8=(EXP(N8)+EXP(-N8))/2
50940 RETURN
50950 REM ** ******************
50960 REM ** COMPUTE TANH N8
50970 N8=(EXP(2*N8)-1)/(EXP(2*N8)+1)
50980 RETURN
50990 REM ** ******************
51000 REM ** DRAW ONE SCREEN LINE
51010 PRINT "--------------------------------------"
51020 RETURN
51030 REM ** ******************
51040 REM ** CONVERT POLAR TO RECTANGULAR
51050 REM ** MODULUS IN M8, ARG IN A8
51060 REM ** REAL IN R8, IMAGINARY IN X8
51070 M8=ABS(M8)
51080 R8=(M8*COS(A8)):X8=(M8*SIN(A8))
51090 RETURN
51100 REM ** ******************
51110 REM ** CONVERT RECTANGULAR TO POLAR
51120 REM ** REAL IN R8, IMAGINARY IN X8
51130 REM ** MODULUS IN M8, ARG IN A8
51140 K8=0
51150 M8=SQR(R8^2+X8^2)
51160 IF R8<0 THEN K8=180
51170 IF R8=0 THEN 51200
51180 A8=((180/PI)*ATN(X8/R8))+K8
51190 GOTO 51220
51200 IF X8>0 THEN A8=90
51210 IF X8<0 THEN A8=270
51220 RETURN
51230 REM ** ******************
51240 REM ** KEY SPACE BAR TO STOP/START
51250 GET K8$:IF K8$="" THEN 51270
51260 GET K8$:IF K8$<>"[SPC]" THEN 51260
51270 RETURN
51280 REM ** ******************
51290 REM ** ROUND N8 TO P8 SIGNIFICANT DIGITS
51300 F8=0:IF N8<0 THEN F8=1
51310 N8=ABS(N8):P8=P8-1
51320 C=INT(LOG(N8)/LOG(10))
51330 N8=10^C*INT(10^P8*N8/10^C+0.5)/(INT(10^P8))
51340 IF F8=1 THEN N8=-N8
51350 RETURN
51360 REM ** ******************
51370 REM ** RANDOM IN R8 BETWEEN
51380 REM ** LIMITS L8 AND H8
51390 R8=INT((H8-L8+1)*RND(1)+L8)
51400 RETURN
```

with high line numbers. BASICs which include the ability to APPEND programs together (or 'Toolkits') normally require the APPENDed code to be at higher line numbers than the already resident code. Thus, it is possible while developing a program to become aware of the need for the subroutines, in which case they can be appended.

USING GENERAL PURPOSE SUBROUTINES

A lazy way and one to which I am addicted, is to first load in the lot before beginning any program. If any are found to be useful then use them; any not used at the end can be erased.

After the program is more or less finished and tested, it may be possible to dispense with pre-assignments in the calling sequence by changing the subroutine variables to suit the calling program direct. It is wise to keep a copy of the pre-doctored version before attempting changes in the parameter names in case you land in a hopeless mess.

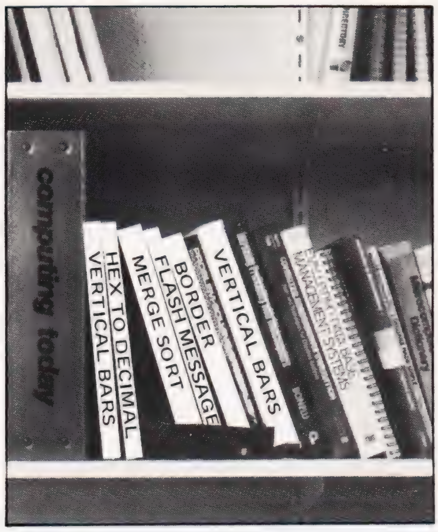
The listing shows the example set of subroutines which occupy lines 50000 to 51410. None of them are revolutionary, most of them lack elegance — but they do work on any PET and, with a few simple modifications, on most other computers. Line 50120 in the 'Print Border' subroutine initialises A8 to the starting address in the screen memory and B8 stores the number of characters in one screen line.

The POKE number 42 is the PET code for the character ' * ' which is used to paint the border. The random number subroutine at line 51390 may need a different keyword to RND(1). The 'Key Space Bar to STOP/START' subroutine at line 51250 may need changing if your GET waits for a character (the PET doesn't).

TESTING LOOPS

To avoid unnecessary clutter, the subroutines have a minimum of validation code. It is, of course, possible to build in traps to keep out absurd input parameters but the normal place for these would be after the INPUT statements in the calling program and it is inefficient to duplicate them again in the subroutine.

However, it is easy to try any of them out first by a simple loop which supplies the required parameters from INPUT. The following is a simple module to test



out the first subroutine at line 50010:

```
100  INPUT "ENTER NUMBER TO BE
     PRINTED";N8
110  INPUT "ENTER NUMBER OF DEC
     PLACES REQUIRED";P8
120  INPUT "ENTER TAB POSITION OF
     DEC POINT";T8
130  GOSUB 50010
140  GOTO 100
```

The method of testing out the others would be similar in principle. The GOTO 100 allows you to try out all possible inputs until you are satisfied with the extent of the limitations. ■

Mike James

ELEGANT PROGRAMMING

It's bug-hunt time as our series takes a look at crashproofing your programs.

So far in the series, we have considered what makes good and elegant programs from a programmer's point of view. In other words, we have been looking at programming as an activity but we have not considered the product in much detail. If you use a logical and ordered approach to programming then the program you produce will be better than if you hadn't. Elegant programming methods do not guarantee a program that a user will admire, only a program that the programmer will admire!

What makes an 'elegant' program from a ***user's*** point of view? The term 'elegant' applied to a program means a wide variety of things to whoever is using it. Clearly how fast and how small a program is are aspects of program efficiency. Just about everything else is a matter of elegance. Two programs may carry out the same job using roughly the same amount of time and memory but deliver entirely different degrees of service to the user. One may 'crash' (ie not complete the job) given only a small amount of user ignorance or perversity, whereas the other may allow the user the liberty of completely ignoring any instructions concerning the 'proper use' of the program.

Other important aspects of elegance are not so obvious. Take, for example, maintenance and extendability. A program may do the job in hand today but what about tomorrow's job? Some programs can be modified easily, others are such a nightmare that it's preferable to start from scratch! These and other aspects of elegance are mainly about using a good programming method and testing the product well, but this is not true of the degree to which a program is crashproof.

A good program is well written and well behaved!

In the early days of computing, the demand was for a program to solve a specific problem and if the result was a program that worked if it was treated with care then everything was fine. Users had to be content with what they got! These days there is no excuse for producing a scrappy program along with a list of dos and don'ts. Indeed the average user demands a program getting the right answer but a program that does not crash no matter what is thrown at it.

One reason for this is that the average user is becoming less technical as computers find their way into domestic applications. This means that programs have to be good due to the intolerance of the non-programmer. We can all imagine a typical scenario — the computer game fanatic kicking his machine to pieces because just before he managed the score of 1,000,000,000 (something only achieved once in a lifetime), the message:

```
***** INTEGER OVERFLOW IN 17839 *****
** SYSTEM ABORT FUNCTION INITIATED **
```

appeared on the graphics screen.

Another reason is that the number and quality of programmers is increasing. This has led to an awareness that things do not have to be quite so bad.

ELEGANCE VS EFFICIENCY

Some programmers would argue that elegant programs are all very well but an efficient inelegant program is much better than an elegant program that takes a week to get its results. This is a valid but misleading argument. It is too often used in order not to bother about elegance. The claim that a better job would have been done if more machine power were available is really no more than an excuse. This is thought to apply to micros in particular because everyone knows that they are not very powerful. However, the following points should be kept in mind:

1) A single user micro can be more powerful than a heavily loaded time share system.
2) The bulk of most conversational programs are not speed critical.
3) The time consuming parts of any program can normally be isolated and treated separately.

As computer hardware becomes cheaper and more powerful, the importance of efficient

programming becomes less and perhaps in the limit, vanishes. In other words:

> ***Efficiency is a hardware problem — programming is about elegance.***

The reason the above maxim is only close to the truth and not entirely true is due to the existence of an area of theoretical computer science/logic known as complexity theory. You can show that for some problems, if you use the wrong algorithm the problem can take as long as the universe has left to it to work out and most algorithms are the wrong algorithm. In this sense, programming is also about finding a good algorithm.

CORRECTNESS, BUGS AND CRASHES

Assuming a program is using a method that will in theory do the job required, there are a number of reasons why it might fail:

1) There may be syntax errors — this is equivalent to a spelling mistake.
2) The method might not be implemented properly, ie the program may not be correct. This is equivalent to not saying what you intended to say.
3) The program might fail because of some user generated condition — this is equivalent to being ambiguous.

The first two are what most programmers will recognise as 'debugging' a problem. The first problem is easy to deal with because your friendly interpreter or compiler will let you know what is wrong. The second problem is very difficult to give any general advice about as it all depends on what alogorithm you *should* be using versus what algorithm you *are* using! You can also find that you are using the wrong algorithm for a wide variety of reasons ranging from just not knowing the correct algorithm to using the wrong variable at some point in a program because of a typing error. The third reason is subtler but a more common reason for a 'fully debugged' program crashing in use. It is worth examining why.

When FORTRAN was introduced as the first high level programming language, it started an important trend — programming by default. For example, the letters I to N were taken by FORTRAN to mean integers unless otherwise instructed. A language such as ALGOL or Pascal would demand that the programmer declare that I,J,K, . . .N were integers (or something) before they were used. Thus, in ALGOL-like languages nothing is assumed about variable names, but in FORTRAN-like languages (including BASIC) variable names are assumed to be one type unless otherwise instructed. The effect of this is that in one case programmers have to build something into their algorithm to deal with variable types and in the other, it can be ignored.

Programming by default seems harmless enough and indeed as long as it's confined to this sort of thing, it is positively an advantage; it saves a lot of time when writing a BASIC program to ignore declaring all the variables before they are used. The trouble with programming by default is that, without realising it, it has managed to work itself into the way an interpreter/compiler implements your algorithm!

ERRORS BY DEFAULT

The most dangerous area of programming by default, because most programmers rely on it, is the automatic definition and handling of run time errors. Look at the error message lists of your BASIC interpreter — each one indicates an error condition that in a good program should never happen. If one of them does occur then the sequence of action which results is usually not under your control. To summarise:

> ***The detection of run time errors and the subsequent actions are usually part of the definition of the interpreter or compiler and are forced on your program.***

To make all this clear, let's consider the innocent looking problem of inputting data. Normally, all that is involved is writing INPUT in a box in a flow diagram and then translating it to an INPUT statement in BASIC. However, INPUT is a very complex operation and there are a large number of ways of interpreting it. For example, what is to be done if the input is of the wrong type? Most interpreters will give a cryptic error message and ask for the input again, some (the worst) simply give the error message and stop! The implementation of INPUT on the PET is particularly frustrating for a beginner — if you type a Return without any input the program stops! This is clearly programming by default!

If we examine how errors come about it's possible to define two types:

1) Inner errors — these are intrinsically difficult to detect within the language being used. For example, it is difficult to discover if arithmetic overflow is going to *occur* before the execution of an expression.
2) Outer errors — these are detectable, in theory, at some point in the program before they occur. For example, in theory, it is possible to check that a file exists on disc before attempting to read it and generating a FILE NOT FOUND type of error.

In an ideal world, the writer of an interpreter would transform all inner errors to outer errors and allow the programmer to try to detect them. Of course, if they are not detected the interpreter must still attempt to deal with them without crashing the program, ie programming by default is not ***all*** bad.

CRASHING IS UNNECESSARY

Now that we have recognised the problem in all its complexity, the solution is obvious. Rather than allow the interpreter to program parts of our algorithm that we haven't bothered about, we must take control! This is easy to say but often very difficult to do. In some cases, it turns out to be impossible because of the poor quality of the BASIC interpreter. There are three broad methods of programming at the level of detail required, however, and some combination of the three should solve most problems.

1) Smaller steps. In the example quoted, programming by default came about because of the different ways in which the INPUT box of the algorithm could be expanded. If, instead of using the blanket term INPUT, we had taken the trouble to define our requirements in detail then (if we are lucky) no crashes should occur. Thus, our first solution is to program ambiguous parts of an algorithm in smaller, more precise ▶

steps. This depends on there being a way of breaking the action down into smaller steps and not all dialects of BASIC will allow this. For example, in PET BASIC (and many others), the statement GET will return a single character typed at the keyboard without waiting for a Return or checking it in any way. (If no key has been typed then a null string "" is returned.) This is obviously the smallest input action that can be defined in common BASIC. Using this simpler instruction, it is possible to build uncrashable input routines. For example, the following short program will read in positive numbers.

```
10  GET A$
20  IF A$="" THEN GOTO 10
30  IF A$=CHR$(13) THEN GOTO 100
40  IF A$<="9" THEN GOTO 70
50  PRINT "DIGITS ONLY PLEASE!"
60  GOTO 10
70  IF A$<"0" THEN GOTO 50
80  I=I*10+VAL(A$)
90  GOTO 10
100 PRINT I
```

Lines 10 and 20 get a character from the keyboard. Line 30 checks to see if Return has been pressed and if it has transferred control to line 100 to print the result. Lines 40 to 70 check that the character typed is a digit and line 50 prints a message if it isn't. If the character is a digit then it is converted to numeric form using the VAL function and added to the running total entered so far in I. Although this program is virtually uncrashable (see later for how to crash it!), it is far from friendly. For one thing it doesn't allow you to edit any number that you enter. Friendly uncrashable input routines can become very long and have to be very comprehensive. (See MAXIMANDER in ***Computing Today,*** Vol 3 No 5, July 1981 for an example.)

2) Use prechecks. It is usually easy to detect an outer error before it happens. The problem is what to do if you succeed in detecting an error. For example, suppose you check the existence of a disc file before trying to read it — what if it's not there? If the program should have created it earlier, the only reasonable conclusion is a machine error — but what if someone has changed the disc during the run or started the program somewhere other than at the beginning. A possible answer is to ask the user if it's a good idea to start again. One thing is certain, however — it is not a good idea to print FILE NOT FOUND and stop the program! If the user was supposed to supply the file name then a polite message suggesting that the user 'have another go at getting it together' is often not enough. I have often been stuck in the middle of a program with the prompt NO SUCH FILE — INPUT FILE NAME AGAIN appearing each time I type another guess at the file name!

ON AN ERROR?

It is possible to pre-check for most inner errors but in general it isn't easy. For example, the input routine given earlier is only almost uncrashable — if you try to enter a number that is larger than the machine cand hold, it *will* crash with a cryptic message from the interpreter. You might think that you can avoid this problem by adding the lines:

```
 75  IF MAXNUM-(I*10+VAL(A$))<0 THEN
     GOTO 110
 :   :                 :              :
 :   :                 :              :
110  PRINT "NUMBER TOO BIG"
120  I=0
130  GOTO 10
```

to the program where MAXNUM is the largest number that the computer can handle. The trouble with this solution is that the program will still crash when the interpreter works out the expression in the IF statement. It doesn't matter even if the whole expression works out to less than the largest number—if you reach the largest number in the course of the calculation then the program crashes! The correct solution is to change line 75 to:

```
75  IF (MAXNUM-VAL(A$))/10<I THEN
    GOTO 110
```

This is the same test but it cannot cause a crash because the expression on the left of the '<' sign is certain to be smaller than MAXNUM.

3) ON ERROR GOTO. Some dialects of BASIC convert a lot of inner errors to outer errors by using an ON ERROR GOTO statement. In BASIC, the ON ERROR GOTO statement is unique in that it doesn't do anything when it is encountered during the running of a program. Following an ON ERROR GOTO 'line number' statement, any error detected by the interpreter causes a GOTO or a GOSUB to that line number. Hence, every error is detectable by the user and is an outer error. (Unfortunately, some BASICs define a set of errors that always cause interpreter controlled handling — this is unnecessary.) Most micro BASICs do have an ON ERROR statement; the notable exception being PET. The way that these work varies quite widely but most supply an additional variable called ERR or ERROR which contain the code of the error that has occurred and a variable called ERL or ERRLINE which contains the line number of the line that the error occured in. Using these two variables it is possible to decide what to do about the error. After the error handling has been completed, control can be passed back to the main program by the RESUME statement which will continue execution at the line the error occured or by RESUME 'line number' which will continue execution at the specified line number.

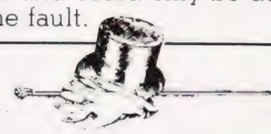
The trouble with using an ON ERROR statement is similar to that of dealing with detected outer errors. A very simple program becomes quite complex if full error handling is included. For example, the very simple addition program:

```
10  INPUT A
20  INPUT B
30  PRINT A+B
40  GOTO 10
```

becomes very long if you use ON ERROR to crash proof it.

```
  5  ON ERROR GOTO 100
 10  INPUT A
 20  INPUT B
 30  PRINT A+B
 40  GOTO 10
100  IF ERR=6 THEN GOTO 200
110  IF ERR=13 THEN GOTO 250
120  IF ERR=23 THEN GOTO 200
130  PRINT "AN ERROR HAS OCCURRED
     WHICH SUGGESTS A HARDWARE
     FAULT"
140  STOP
200  PRINT "THE NUMBERS ARE TOO BIG
     TO CARRY OUT THE ARITHMETIC"
210  IF ERL=30 THEN RESUME 10
220  RESUME
250  PRINT "PLEASE TYPE A NUMBER -
     NOT LETTERS"
260  RESUME
```

If any error occurs while the program is running then control is transferred to line 100. The variable ERR is checked to see what error has occured and various messages are generated to inform the user. (To explain the error codes — 6 is overflow, 13 is type mismatch and 23 is linebuffer overflow.) Notice that the action taken depends on the error code and the line number that the error happened in. Also, the only condition that causes the program to stop is when the error is totally inexplicable and could only be due to a machine fault.



If a BASIC has an ON ERROR statement then there is no excuse for any interpreter generated error messages.

Next month we move away from programming techniques to look at some very necessary background information — randomness and its use in programs. ■

PROGRAM POWER MICRO POWER

Attention all BBC & ATOM Owners !!

Acorn ATOM
MACHINE CODE PROGRAMS

NEW **Dog Fight(12K)**
Exciting, two player combat program – in HI RES Graphics. Defend your aircraft carriers and attack your opponents. Engage the enemy fighters in a battle to the death. £5.95

NEW **Stock–Car(12K)**
HI RES, Two Player contest around any of 16 different race tracks. Steering, acceleration and braking controls. Set the level of difficulty for driver and choose the no. of laps. £5.95

NEW **Astro Warrior(12K)**
The ASTEROIDS IDEA, but developed and improved !! Flying around the screen you must shoot your way out of difficulty. Aliens attack in increasing numbers. Refuel at specified fuel dumps before you run out. A HI RES game for experts. £5.95

NEW **Super Breakout(3K plus 3Kgr plus Fl. Pt.)**
NOT just ANOTHER version !! Equally good in colour and black and white, the program has the advancing wall(s) feature plus genuine curved bat action with 6 ball angles. Terrific sound effects make it a MUST for arcade game addicts. £4.95

NEW **Swarm(12K)**
Alien Killer Bees are swarming the Earth. The Queen Bee is indestructible. Your only hope is to destroy the grubs and new born males before they wipe you out. You get three lives with a bonus at 10000 points. (HI RES graphics). £4.95

ROM SELECTOR BOARD
Suffering from congestion? (PROGRAMMER'S TOOLBOX WORD PROCESSOR etc.) Switch between up to 4 ROMS located at Hex A000. Assembled and tested unit with compact professionally produced PCB and good quality components. Fits easily into the Utility Socket (IC 24). **Only £19.50**

Adventure
ALL THE EXCITING, INTRIGUE AND FRUSTRATION OF A MAIN–FRAME ADVENTURE! 12K
Explore the tortuous forests, dark caverns and castle dungeons. Beware the maze of twisting tunnels and the desert wastelands. Outwit the predators. Rescue the PRINCESS and carry off the treasures.
Great Skill and imagination are required to play this excellent game and you may still never exhaust all the possibilities.
By devising methods of condensing messages, the author has been able to include many features which would otherwise be available only on much larger computers.
Start your adventure now – £7.95

Chess(12K)
Improved graphics, plays black or white, mid game level changes, look ahead up to eight moves, offensive, normal and defensive play. Ten sub–levels, castling, 'En passant' by player. Rejects illegal moves. Take back moves and action replay with take over. Set up problem games. £7.95

Cowboy Shoot–Out(12K)
Full feature, two player, arcade shooting game. Cactus plants, wagons, animated cowboys. Superb graphics and sound. £6.95

Hyperfire(12K)
Aliens have over–run six planets. Object to clear them. Super high–speed action. Four defender ships. Hit alien fuel dumps and alien ships. Aircraft radar shows placement of approaching ships. Thruster control – including diagonal movement. Best sound effects yet! Avoid shots from outside vision scan £7.95

Air Attack (12K)
Command an East Coast anti–aircraft battery. Destroy enemy fighters, bombers and doodle–bugs. Ingenious graphics depict planes approaching, veering and flying off. Search the sky and scan the landscape through 360 degs. GOOD SOUND. GREAT ENTERTAINMENT. £7.95

Space Fighter(6K 3Kgr)
Super High Speed "Defender" game. Five types of intelligent aliens. Repeating laser cannon, smart bombs, hi–score, rankings, bonus points, six skill levels. Exciting sound effects. £7.95

BBC MICRO

NEW **MODEL B (or 32K MODEL A)**
Allen Destroyers
Sensational, high speed program with an abundance of features. Brilliant use of colour graphics and sound. 48 strong Invader Fleet of three different types plus Mothership scoring mystery bonus. Choice of six invader speeds and three bomb speeds. Vertical, angled and exploding missiles. Options to replace defences and suppress new fleet advance. Bonus bases awarded each new sheet. Scoring according to overall difficulty level. Ongoing display of score and hi–score and end of game ranking of top five scores.
This program has many unique extras e.g. a 'battle analysis' showing number of each invader shot down, how many Motherships destroyed, number of sheets cleared, shots fired, percentage hits and bases lost.
If you needed a reason to upgrade to 32K, now you have it! £6.95

Timetrek
The ultimate 'real–time' Startrek, with Battle Zone indecision your main enemy. Brilliant colour and sound. 'One–screen' presentation. Twenty skill levels. Panic Button for 'once–only' space leap. Torpedo sight control. A programming masterpiece!! £7.95

NEW **Eldorado Gold**
Welcome to a past age, when cowboys ruled the roost and Indians terrorised the new settlers. In ELDORADO COUNTY many fortunes were made – and nearly as many lost! Legend has it that Old Bill McCluskey, who met a rather sudden death, had built up a vast treasure somewhere in the nearby territory. Can you end up a rich man where many have failed? If the marauding Apaches or Big Jake's gang don't get you, you may die of thirst in the desert or starve to death in jail. £6.95

World Geography
Beautifully drawn HI RES colour map and graded testing of world capitals and populations, makes learning geographical facts a pleasure. 160 countries covered in total. Pin–pointing of each location in turn aids general awareness. £5.95

NEW **Space Maze**
You have crash landed in the legendary labyrinth of Titan, inhabited by alien monsters known as 'Froogs'. Find your way out to the 'Transmat' probe before being cornered and eaten. Eight skill levels and 3D colour graphics. £5.95

Astro Navigator
Navigate your way through a variety of treacherous caverns, inhabited by killer rockmites. Avoid the dangers with your climb, dive, reverse and thruster controls. 5 Skill Levels. Top 5 scores ranked. Excellent colour graphics and sound in this exciting 'real–time' game. £4.95

NEW **MODEL A or B**
Chess
High quality chess game with castling, 'en passant', play black or white, rejection of illegal moves and six levels of play. Set up problem games. £7.95

Micro – Budget
Proven personal finance program which enables you to record, review, analyse and budget your income and expenses for twelve separate monthly periods. Coding system gives sub–totals, selective summary of entries and cumulative cash flow. All figures can be saved to tape. £6.95

Munchyman
Colourful and highly entertaining version of this popular arcade game. Munch your way to a high score, before the 'munchers' devour you. Reverse roles by munching the stars. The more you score, the harder the game gets. £5.95

Other programs available:
Startrek £5.95/Gomoku £3.95/Disassembler £5.95
Cat & Mouse £4.95/Zombies (New) £3.95

APPROVED ACORN AND BBC STOCKISTS

1250 ATOM USERS CAN'T BE WRONG!

PROGRAMMER'S TOOLBOX
(PACKED 4K EPROM) £24.50
★ 1200 BAUD CASSETTE OPERATING SYSTEM
★ VISIBLE LOAD & SAVE

● TRACE (X)	READ	ON ERROR	HEX
● LTRACE	DATA	RENUMBER X, Y	IHEX
● STEP	RESTORE	AUTO X, Y	OFF
FIND	ELSE	CURSOR X, Y	VECTOR
VAR	WHILE	BEEP X, Y	ZERO
LVAR	ENDWHILE	KEY X	POP
DELETE X, Y	DUMP	INKEY $X	STOP

(● VIA chip required)
Suitable for any memory size
Greatly enhances ATOM existing facilities

WE *Guarantee* THAT ALL OUR ADVERTISED PROGRAMS HAVE BEEN COMPLETED AND ARE READILY AVAILABLE

SPECIAL OFFER
Deduct £1 per cassette when ordering two or more.

MICRO POWER LTD.
8/8A REGENT STREET,
CHAPEL ALLERTON
LEEDS LS7 4PE.
Tel. [0532] 683186

Please add 55p order P & P + VAT at 15%

Please Note:
All programs are now available at all good dealers or direct from MICRO POWER LTD.

PROGRAM POWER MICRO POWER

PRINTOUT

Dear Sir,

In response to the many queries which, I understand, your magazine has received, I would like to personally explain the current delivery situation for our new ZX Spectrum personal computer.

The general public's response to our new computer has far exceeded our expectations and we have been 'swamped' with orders. This and some small initial production delays have led in turn to considerable delays in delivery.

Regrettably, many of our customers may have to wait up to 12 weeks, from our receipt of their order, for delivery of their Spectrum and we are writing to them all to apologise for the inconvenience and to offer them the chance of an immediate refund.

For those customers who continue to wait, we shall be sending out with each Spectrum, in compensation for the delay, a £10 voucher, which can be used in part-payment for a ZX Printer or to buy a complete pack of five rolls of printer paper.

We are also providing customers with a new demonstration tape containing:
1. A complete 'keyboard trainer' to introduce the Spectrum,
2. Three major programs — an exciting game, Through the Wall, a drafting program, Draw, and Character Generator, which demonstrates user defined graphics,
3. A series of illustrative programs — Bubble Sort, Evolution, Life, Monte Carlo and Waves.

Finally, I would like to assure you and all our customers that the initial problems with the Spectrum have now been completely overcome. Production is running smoothly at 5,000 units per week and will rise sharply over the coming months. We are confident that our present backlog will be cleared by the end of September and hope that you will see current delays in the context of our successful delivery of more than 500,000 computers in the last two years.

Yours faithfully,
Clive Sinclair
Sinclair Research Ltd
London SW1X 8LB

(*The latest news from the Sinclair stable is that the price of the ZX81 has dropped from £69.95 to £49.95. Sinclair have also signed Greens, a subsidiary of Debenhams, and Boots as high street retailers for the ZX81 range; rumour has it that two futher high street retailers will be added to the list over the next month as well. Ed.*)

Dear Sir,

I was most interested to read Nigel Searle's defense of the Spectrum advertising stating that the Spectrum exceeded the BBC Model A Micro on specification and price. It's a pity Mr Searle couldn't answer the questions directly instead of quoting two 'independent sources'. I would hardly have thought that **Sinclair User** *dedicated as it is to the Sinclair machines could be termed 'independent'. As for Tim Hartnell's review for* **Your Computer**, *there was no evidence to back up the conclusion in the review that the Spectrum "exceeds the BBC Microcomputer Model A". I wrote to Tim Hartnell myself asking him to justify his claims and received a reply (on notepaper headed ZX User's Club) that stated "What I said in my review as submitted to the magazine was "on paper the Spectrum specifications exceed those of the BBC machine". What I meant to imply was that specifications on paper were not the whole story. The BBC machine will be considered a good machine in two or three or more years. The Spectrum is an ephemeral computer".*

It's a great pity that Sinclair seem to have to criticise the opposition to sell their machines rather than letting the machine sell itself. If I may use Mr Searle's example of quoting other people rather than justifying claims myself, Guy Kewney described the Spectrum in **Which Micro?** *magazine, thus: "I was left feeling unnecessarily disappointed by a nice enough machine. Most of this I blame on having been promised a machine that was better than the BBC machine. I think the BBC machine has something extra which makes it worth the extra money". Mr Kewney does not have any interest in either Sinclair or BBC associated user groups or publications!*

Yours faithfully,
I D Smith
Eastleigh

Dear Sir,

*Mr Beattie, in his letter (***CT** *Aug) denigrating the Spectrum, sounds like someone trying to justify the extra £200+ he has paid to acquire the BBC 'B' Micro. Methinks he doth protest too much!*

Sinclair, in reply, used two review extracts. I have read many more and have yet to find one that does not use the price edge to more than justify the purchase of a Spectrum. No way can there be sense in paying more than double (basic Spectrum to basic BBC) to acquire a machine that barely equals the Spectrum. Indeed, were it not for the edge the BBC machine has with its keyboard, running speed and, as one reviewer put it, "it looks more like a micro should . . ." there would not be a contest.

The Spectrum may well have the promised fight on its hands with the new VIC and ELECTRON but unless they are priced right and can also compete in due course with the promised ZX Microdrive at "around £60" they are also on a hiding to nothing. Price isn't everything, but in today's economic climate it is the biggest single influence and only a fool would ignore it.

Yours faithfully,
R W Spiers
Whitstable

(* The two letters printed above represent both sides of the argument sparked off by Mr Beattie's letter in the August issue. Judging by the vast quantities of post, some 30-40 letters to date, it would appear that the balance is firmly on the side of Mr Beattie. However, I think the time has come to say STOP, simply because the volume of mail on a single subject tends to swamp the various other views that people want to express. Ed. *)

Dear Sir,

A quick tip for the Spectrum owners among your readership.

The Spectrum manual emphasises that programs must be saved with the EAR jack plug disconnected. This is a very irksome (and necessary!) procedure, and one that causes a lot of plugging and unplugging when a program is being developed, in addition to a number of lost programs. I found, ▶

however, that one of my cassette recorders (a Trophy CR100, from Currys) did not need this unplugging. This was a recorder on which I had soldered a 330R resistor between the output to the EAR jack and the loudspeaker terminal. Normally this connection is broken when the EAR jack plug is inserted, so that the speaker is inactive. Since I like to hear my prorams loading (a hangover from TRS-80 days), I connected the resistor so that the speaker would output at low level as the program loaded.

I imagine that the 330R resistor is enough to remove most of the feedback signal which affects the SAVE operation of Spectrum. It's almost certain (I haven't tried it this way round) that the same value of resistor connected between the terminals of the EAR jack plug itself (so as not be disturb the recorder) would also allow you to keep both leads plugged in during SAVEing and LOADing, so speeding up your Spectrum programming. This would involve cutting off the original plug, which is moulded, and replacing it with another — the 330R resistor could be incorporated inside the new plug if a miniature resistor is used. Perhaps some hardware supplier will market new leads!

Yours faithfully,
Ian Sinclair
Halstead

Dear Sir,

I have to admit to a mistake in my BBC Programming article in July with regard to the memory map of the BBC Micro. My explanation of the memory map is correct except for the details of the way individual bits control the colour displayed in a four- or 16-colour mode. I assumed that the bits relating to each pixel (picture element) would be grouped together. The program that I used to test this assumption seemed to confirm this very logical way of storing the information. Unfortunately, I used a black and white set to view the results and being convinced that logic is always right I failed to assess the input from my senses! It is not often that I can claim to have learned so much from one mistake! As a result of reader's letters I can now not only set the record straight but say that I enjoyed reading what other BBC Micro owners are finding out about their machines. There is no doubt that the BBC Micro has an unusually knowledgeable and enthusiastic group of users and I hope that Acorn realise that there is interest in the more technical aspects of their machine and publish more data!

The correction to the article is basically a change to Fig. 4. The bits corresponding to a single pixel are split between the pair of 'nibbles' (four bits) such that the first bit of each nibble determines the colour of the first pixel, the second bit of each nibble determines the second pixel and so on. This odd storage pattern is probably used to increase the speed that data is transferred to the video ULA. Whatever the reason, it certainly complicates matters and makes PEEK and POKE graphics in anything but a two-colour mode difficult to do from BASIC. This is not to say that knowing the correct memory map is unimportant because it is essential if you want to write a screen-to-printer dump program.

I would like to thank in particular Mr T Mabbs, Mr P K Chilvers and Mr J N Nell for their very helpful comments and information. I also received, courtesy of the Editor, a 'free style' postcard in dayglow colours from Mr R Gordon, suggesting that I should be sacked (from what he doesn't say!) and turn in my BBC for committing the crime of yet another error — that of leaving a '?' out of the definition of the FNS function in the same article. Now, I will admit to one error a month but two . . .never! The function is correct as written and returns the address of a screen location as described in the article. It can be used with a '?' in front to return the content of a screen location — ie FNS returns an address and ?FNS returns the contents of the address. The point is that sometimes you need to look at the address and sometimes the content so building the '?' into the function isn't a good idea.

I rest my case . . .but not my BBC Micro!

Yours faithfully,
Mike James

Dear Sir,

I wonder if I might comment briefly on your review of word processing books in your August 1982 issue — especially the fact that your reviewer believed that the **Textedit** *book was the first of its type.*

It is a very useful book but by no means the first one. There have been many books which consist of a program listing of a single topic — perhaps the most famous is the Sargon chess program for which a Z80 assembler language program was published by Hayden Books in 1978.

If we look at books of collected programs, then there are many hundreds of these, many of which provide very useful programs on a wide variety of engineering, scientific and business topics (as well as games).

The difficulty is, of course, in knowing what the books contain, so that if you are wanting a word processing or data base program you know where to look. The answer to this problem is the **Small Computer Program Index** *which provides a reference to printed program listings from both books and magazines. Over 2,000 programs will be indexed in 1982 and the cost for all this information is only £15 (about half the cost of a box of discs!). Further information can be obtained from ALLM Books, 21 Beechcroft Road, Bushey, Herts WD2 2JU.*

Yours faithfully,
Alan Pritchard
(Editor, SCPI).

(* To add a quick rider to the above, I think it only fair to say that we actually meant that it was the first DIY word processing package and not the first DIY package *per se*. Ed. *)

Dear Sir,

I have been a reader of your magazine since mid-1980, when I took up computing as a retirement hobby. I take several other magazines (which shall be nameless) but still prefer **CT**.

I regard myself as a reasonably intelligent person, and have reached a certain degree of proficiency in the use of the computer.

Having established that I am not an absolute beginner, nor an absolute moron, I come to the reason of my retreat from the computer to the typewriter. I have a complaint to make about the articles, or rather the writers of the articles, which are published. Namely, why do they assume that the persons who will read their articles are in the Clive Sinclair class????

My present outburst has been prompted by the article in this month's magazine re; sending TRS-80 graphics to a graphics printer.

I use a TRS-80 Model 1 with discs and have just purchased a Seikosha GP 100A because it had graphics capabilities, but I am still struggling to convert the manual (such as it is) into something I can

understand. I have now reached the point where I can produce a form of graphics, but the means of doing so is diabolical!

The retailer of the machine has not been able to help nor the main distributors. Unfortunately the distributors are in Bristol and the retailer is in Leicester and my 'phone bill will keep British Telecom going for years.

I digress. The writer of the article goes on at great length about seven bits and eight bits, etc, then goes on about jumping into ROM and jumping out again, taking up two and a half pages in the process. He is evidently a learned person, and I give him credit for that, but what these writers and I am afraid Editors of magazines seem to overlook, is that very probably the readers of the magazine have not necessarily their expertise.

Anyway, this seems to be turning more into an article than a letter so I will close with a suggestion. Could you not have someone on your staff who is either a non-expert, or who could alternatively play the role of a non-expert, and keep saying to these boffins 'WHY'??? Possibly then they would go back to basics (forgive the pun) and explain their ideas 'as to a child.'

Yours faithfully,
C Furness
Sheffield

* I don't think your problem really lies with the content of the magazine, although I will admit that it is not really written for beginners, but with the writers of Japanese manuals. The Seikosha is supported by a particularly awful piece of documentation but, as is so often the case, there are better descriptions of its functions available through some of the OEM users. So, if you bought a Seikosha in someone else's box you might stand a chance of getting it to perform . . . silly but true. Perhaps there's the makings of an article here . . . watch this space. Ed. *)

Dear Sir,

The following reveals that the Sinclair Spectrum is capable of producing up to (a theoretical) 128 colours.

How many colours in a Spectrum? Seven or eight, if you accept Sinclair's inclusion of Black and White as colours. But how about 64? Or 128? Well, it is possible for the Sinclair Spectrum to produce up to 128 different hues. This is done by creating a graphic character resembling a chess board (as on the 'A' key of the ZX81) so that the basic colours may be mixed at the level of high resolution pixels. My program uses such a graphic character, varying the INK and PAPER across all permutations, to produce 64 hues (there are eight colours for each of the INK and PAPER). Another 64 can be obtained by varying BRIGHT between 0 and 1 (line 60). The program prints out the PAPER and INK values of both for reference and to give examples of what text overwritten on these colours is like. Clearly, 128 distinct colours are not possible, but the range is much larger than just the standard six colours plus Black and White. The second part shows the colours a page at a time.

Yours faithfully,
Tim Langdell
West Dulwich

```
  5 REM ** Checkered colours. NOTE
    that [n G^ A] is 'n graphics
    characters obtained from key A'
 10 FOR a=0 TO 6 STEP 2
 20 POKE USR "a"+a, BIN 01010101:
    POKE USR "a"+a+1, BIN 10101010
 30 NEXT a
 40 FOR p=0 TO 7
 50 FOR i=0 TO 7
 60 FOR b=0 TO 1:PRINT PAPER p;
    INK i; BRIGHT b;"[2 G^ A]";p;
    i;:NEXT b
 70 NEXT i:NEXT p
 80 REM ** An even better example
 90 FOR p=0 TO 7:FOR i=0 TO 7:
    FOR b=0 TO 1
100 FOR y=0 TO 31:PRINT PAPER p;
    INK i;BRIGHT b;"[33 G^ A]";:
    NEXT y
110 POKE 23962,255:REM ** Scroll
    automatically
120 NEXT b:NEXT i:NEXT p
```

Dear Sir,

I was pleased to see T P Mervyn's program for the date of Easter in August's **Computing Today.** *Unfortunately he did not research his subject adequately for, in using the Clavius/Lilius algorithm, he did not take into account the fact that Britain did not follow Europe in bringing the calendar into alignment with the sun until 13th September 1752. So the program will not work for the UK prior to that date.*

Easter began to be celebrated by Christians about AD 60. Until about AD 300 most Churches celebrated Easter at the time of Jewish Passover, although some isolated groups, including the Celtic Church of Northern Britain, had their own ways of calculating when Easter should be. About AD 300 the Bishop of Rome started fixing Easter on an annual basis but that proved unsatisfactory, so in AD 325 the Council held at Nicaea fixed Easter on the first Sunday after the first full moon after the Spring Equinox.

However, the calendar was not very accurate in those days for the year was taken as being 365 days 6 hours long. In fact the year is 365 days five hours 48 minutes 46 seconds long. They were gaining one day in every 128 years on the sun. Roger Bacon in AD 1263 suggested to the Pope that the calendar ought to be altered and brought in time with the sun but nothing was done until AD 1582 which was the date given by Mr Mervyn in his article.

By that date, the Reformation had taken place and the Protestants were not going to be told by Pope Gregory XIII how to adjust their calendar, so for over a hundred years there were two calendars being used in Europe. In 1700, the Dutch Protestants altered their calendar to make it conform with the sun's position, but Britain held on to the old style until 1752. In that year September 3rd became September 14th. The change caused riots all over Britain with mobs shouting, "Give us back our eleven days!". So, to find the date of Easter prior to 1752 the eleven days have to be incorporated into the program. Two lines only need to be added.

```
215 IF Y<1753 THEN E=E+8
365 IF Y<1753 THEN N=N+3
```

I use a Texas Instruments 99/4A computer myself and I enclose the whole program written for Texas Extended BASIC, a really splendid version of BASIC. I think this should be accurate right back to 325 AD but I have only checked it for the year 1738 by using John Wesley's Journal to find the actual date of Easter for that year, and for the year AD 729 by using Bede's **'History of the English Church and People'** *(Book V: Chapter 22) for the death of Egbert at Iona on Easter Day.*

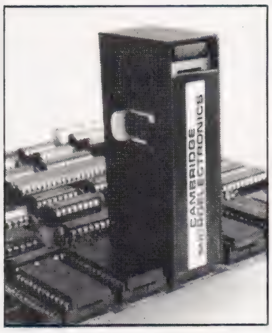
Yours faithfully,
Rev. C D Blount BA FRAS,
Wakefield

```
150 CALL CLEAR
160 INPUT "YEAR FOR WHICH EASTER IS
    REQUIRED?":Y
170 G=Y
180 IF G<19 THEN 210
190 G=G-19
200 GOTO 180
210 G=G+1
220 C=INT(Y/100)+1
230 X=INT(C*3/4)-12
240 Z=INT((8*C+5)/25)-5
250 D=INT(5*Y/4)-X-10
260 E=11*G+20+Z-X
270 IF Y<1753 THEN E=E+8
280 IF E<0 THEN E=E+30
290 IF E<30 THEN 320
300 E=E-30
310 GOTO 290
320 IF E=24 THEN 350
330 IF E=25 AND G>11 THEN 350
340 GOTO 360
350 E=E+1
360 N=44-E
370 IF N<21 THEN N=N+30
380 B=D+N
390 IF B<7 THEN 420
400 B=B-7
410 GOTO 390
420 N=N+7-B
430 IF Y<1753 THEN N=N+3
440 IF N>31 THEN 470
450 CALL CLEAR::PRINT "EASTER DAY IS
    MARCH";N;Y
460 STOP
470 CALL CLEAR::PRINT "EASTER DAY IS
    APRIL";N-31;Y
480 END
```

★ NEW FROM CAMEL PRODUCTS ★

MEMIC T1618 L1424

2kB CMOS RAM UNITS



- Towerblock model with NiCd batteries for high density layouts
- Low Profile cabled version with Lithium battery for use on closely stacked daughter boards
- Useable as a RAM in RAM/ROM/EPROM sockets
- ONLY **£29.95** VAT incl.
- MEMIC-81 Specifically for ZX-81, with rear connector extension **£24.95 + VAT**
- MEMIC-81.2. 4kB version **£29.95 + VAT**
- PIO-81 8+8 Channel latched Input/Output card for ZX81 **£13.00 + VAT**
- ROM-81 A 2kB-8kB ROM/EPROM Unit for ZX81 **£14.95 + VAT**
- PAD 1 An Apple Slot connector card with 12 inch colour coded cables for external connections **£5.17 + VAT**
- PRINTER STANDS Standard and Large for fanfold paper, forms, labels etc., under Epson, Microline, Sheikosha etc., printers. ¼" tinted Perspex.
  PSS £15.95 + VAT
  PSL £17.95 + VAT

CAMEL PRODUCTS

Cambridge Microelectronics Ltd, One Milton Rd, Cambridge. CB4 1UY Tel (0223) 314 814

THE '6809' CENTRE

The FLEX9 Operating System for '6809' Based Micro-Computers.

This disk operating system is now the standard single user system for machines using the powerful '6809' micro.

Having many advantages over similar operating systems for other micro-computers, this system can now be run on many different '6809' based machines. Furthermore, the Configurable Version is available for implementation on custom '6809' computers. Programs written to run with this operating system are almost completely machine independant, thus allowing the easy transfer of software. There is a wide range of software available for this system, some of which is listed below. Please send for our full list

	Excl VAT	Inc VAT
DRAGON 32,6809 based 32k colour computer	173.04	**199.00**
'6809' HARDWARE, RUNNING FLEX9.		
Convert your APPLE II to FLEX – with the new EXCEL-9 '6809' board Complete with FLEX9 on 5¼" disk.	181.74	**209.00**
APPLE II, with EXCEL-9 Board, 48K RAM, 40 x 24 Video Output, two 5¼" disk drives, with 240 Kbyte capacity, with FLEX9	1549.57	**1782.00**
FUJITSU MICRO-8, with 64K RAM, colour graphics on 14" screen, two 5¼" disk drives with 656 Kbyte capacity, with FLEX9	2346.09	**2698.00**
SWTPC S/09, with 64K RAM, 12" intelligent terminal, two 5¼" disk drives with 740 Kbyte capacity, with FLEX9	3338.26	**3839.00**
STIRLING 77-68 KIT, based on 8" x 8" bare PCB's, now with 64K RAM, and Disk Controller for 5¼" or 8" Disks.		

Send for further details on any of these systems, or their component parts.

'6809' SOFTWARE, RUNNING UNDER FLEX9, on 5¼" or 8" disks.				
Configurable Flex, with Text Editor & Assembler	120.00	**138.00**	120.00	**138.00**
Soft/Merge Package, with File Editing Facilities	48.70	**56.00**	58.26	**67.00**
Debug Package, a complete 6809 Simulator	48.70	**56.00**	58.26	**67.00**
Disk Utilities, Additional Commands for FLEX9	48.70	**56.00**	46.96	**54.00**
Text Processor, for Text Formatting	48.70	**56.00**	58.26	**67.00**
Text Editor & 6800/6809 Macro Assembler	69.57	**80.00**	69.57	**80.00**
Disk & Memory Diagnostics, Further FLEX9 Commands	48.70	**56.00**	52.17	**60.00**
Extended Basic, Advanced Business Basic for 6809	62.61	**72.00**	75.65	**87.00**
Pascal, 6809 Machine Code Compiler	130.43	**150.00**	145.22	**167.00**
RMS, A Record Management System for 6809	145.22	**167.00**	145.22	**167.00**
Stylograph, The Complete Word-Processing Package	173.04	**199.00**	173.04	**199.00**
Micro-Modeller, Now Available for FLEX9	694.78	**799.00**	694.78	**799.00**

'6800' & '6809' Cassette Software also available

✻ MAIN LONDON DEALER FOR SHARP AND APPLE PRODUCTS ✻ Come and try any of our systems or make use of our mail order service ✻ All prices include carriage charges within the mainland UK ✻ All prices are correct at time of going to press and subject to availability ✻ Send for our latest price list and catalogue covering our vast range ✻ Further information is available on all the products listed, send for details ✻ Store opening hours – 9.30 to 5.30 – Monday to Saturday –

STIRLING MICROSYSTEMS LTD.,
The '6809' Hardware and Software Store. Ring now for new low prices.

STIRLING MICROSYSTEMS

241 Baker Street, London NW1 6XE. Telephone: 01-486 7671.

T1-99/4A • VIC 16K • ATARI 400/800 • SHARP MZ80k/A/B • BBC MICRO • SINCLAIR SPECTRUM

Buy two at £19.95 — take 1 at £5.95 FREE!!

DATABASE
The program that everyone needs. Facilities include sort, search, list, delete, change, totals, save file, line print if required, etc. etc. Can be used in place of any card index application. **£19.95**

STOCK CONTROL
All the necessary for keeping a control of stock. Routines include stock set up, user reference no., minimum stock level, financial summary, line print records, quick stock summary, add stock, delete/change, record, and more. **£19.95**

MAILING LIST
A superb dedicated database to allow for manipulation of names & addresses & other data, with selective printing to line printer. Features include the facility to find a name or detail when only part of that detail is known. Will print labels in a variety of user-specified formats. **£19.95**

DECISION MAKER
A serious program that enables the computer to make a sound decision for you based on various criteria. If you want to buy a car, hi-fi, house, etc., or you don't know which woman to marry then you need this one. **£5.95**

INVOICES AND STATEMENTS
Ideal for the small business. A complete suite of programs together with generated customer file for producing crisp and efficient business invoices and monthly statements on your line printer. All calculations including VAT automatic and provision for your own messages on the form produced. Not Spectrum. **£19.95**

RUBIK SOLVER
It's not our policy to offer games but we make an exception here for a program to solve the cube from ANY position. Shorthand notation makes learning the solution by heart possible for most active brains **£12.95**

COMMERCIAL ACCOUNTS
A gem of a program, all for cassette, with the following features:—

Daily Journal	Credit Sales	Cash Sales
Credit Purchases	Purchases — other	Sales Ledger
Purchase Ledger	Bank Account	Year to Date Summary

A fully interactive program suitable for all businesses. Files can be saved and loaded and totals from one file carried forward to another on cassette. Particularly useful from a cash flow point of view, with an immediate accessibility to totals for debtors and creditors. Bank totally supported with entries for cheque numbers, credits and, of course, running balance. **£19.95**

HOME ACCOUNTS
Runs a complete home finance package for you with every facility necessary for keeping a track of regular and other expenses, bank account, mortgage, H.P. etc etc etc!! You'll wonder how you ever managed without it. **£19.95**

CHOPIN - LES ETUDES
Six beautiful studies from Opus 10 and 25 performed for you live by Mr. Sharp — Spellbinding!!! **£5.95**

MOTOR ACCOUNTANT
Find out exactly what that car is costing you and keep a data file with all your expenses therein! **£5.95**

RECIPE FILE
Let all those computer widows have a bash! The wives will really enjoy the fun of this program designed to keep all her recipes. Will even suggest a menu for a day/week! Excellent value. **£19.95**

Access Welcome

Gemini MARKETING LIMITED

9 Salterton Road, Exmouth, Devon EX8 2BR Telephone: (039-52) 5832

All prices include CAT and post & packing except Hardware. Full range of Sharp peripherals available. Full range of printers and interfaces at discount prices. Please 'phone for quotations. Please state machine type when ordering software.

STOP PRESS! BBC WORD PRO £19.95!!

Henry Budgett

#FILE

The Editor takes time off this month, not to skip writing his column but to see a new movie.

What's this, you say. The Editor's taking up rather a lot of space this month . . .two pages indeed, he'll be wanting the whole magazine next! Well, it can hardly have escaped your notice that this month's magazine looks just a little different from the one you were reading last month and this bumper piece of editorial is partly as a result of those changes and partly because I wanted to bring you news of a new film that'll be on show in your High Street later this year.

First, however, let's take a look at the changes we've made to the magazine. Well, as far as the editorial material goes, it's strictly business as usual which is good news. Indeed, the main reasons for the change are simply to bring the magazine visually up to date rather than to alter its contents and direction. At this time of the year, it is traditional for magazines to do a little bit of promotion and we usually slip in a few special features. . .this year is certainly no exception! We've packed in a complete spreadsheet program, a load of good programs, a couple of reviews and as if that was not enough, we're giving (yes, giving) away a computer book in three monthly instalments as well as a load of other goodies. We are also going to keep you on your toes with a competition which has a complete Spectrum system (48K computer plus printer plus software) as its first prize and it's all down to skill — none of those silly little 'write a ten word slogan' ideas here.

If you've got any comments to make about the new style, good or bad, please write and let us know. And talking of writing, I must mention the amount of mail which is still coming in over the late delivery of BBC Micros — we've even had one from Australia! Now, I know it's a drag waiting up to a year for the thing and we all agree that the problem should have been solved a long time ago (even some polite and sensible letters from various companies might have helped) but please don't write to me! There is, sadly, very little we can do to help except pass your letters on to BL Marketing, Acorn or the BBC. As far as we know the signs for Model Bs are still not good although Model As should be down to a four week delivery, and the indications are that unless drastic action is taken by the BBC and Acorn, little change is likely for several months to come.

GOSSIP CORNER

Among the more interesting snippets gleaned this month are the following which I thought you might like to share. The boys down at the Commodore offices in Slough are certainly burning the midnight oil these days. Strong indications are that they'll be doing the dirty deed to just about every system under £300, and to a few priced above, with their new Commodore 64 (or rather the VIC 40 as it seems to be called these days). Quiet words in various people's ears seem to confirm that the product will be selling sub £300 and possibly as low as £250 if the market will stand it. The machine has been enhanced since its first showings on that momentous Hanover trip (come back, brain, all is forgiven) and is due to roll in November. Also scheduled for that time is the VIC 20's baby brother, the VIC 10, and this should be software compatible, in the upwards direction, with the 64/40. The 500 series is also expected about then and will be carrying a secret weapon which may well turn out to be either an inbuilt modem or a Prestel unit.

At a recent Press talk, Apple somewhat unexpectedly produced figures which showed that they were second in the league tables for Europe to the big C — honesty, I like! Information drifting in on the Apple front indicates that since they added Keith Hall to the team (ex of Commodore) their dealer network has been tidied up somewhat considerably. While this is probably no bad thing in some cases, I wonder if it reflects a little pique at the fact that Xerox appear to have dropped the Apple system from their High Street range in the States? Further intelligence also points to the possibility of Apple II being replaced shortly — no surprise really — but the whispers also say that the Apple III might be for the breakers yard as well.

HP ADDS ONE

The really interesting news at the 'quality' end of the micro market was that HP have added another machine to their HP-80 range: the HP-86. Looking rather like the '83 but without the integral monitor, it offers a 48K BASIC, 64K

Right: David Warner doing a quick burst of callisthenics in front of the image of the MCP.

Below left: Tron and his lady taking a stroll through one of the more peaceful sections of the film.

of RAM and all the peripheral interfaces built in for some £1,250. Now, while that's expensive in some people's minds, the real game is given away by the fact that the extras are really cheap, by HP's standards. The monitor will set you back about £200 to £230, a disc drive will add about £600 and the wonderfully named Sweetlips plotter comes in at about £1,000. Have HP finally started to change their pricing policy, one asks? If they have, and it certainly looks that way, life in the professional micro market could get very interesting indeed because if HP can cut costs, so can an awful lot of others like ACT, IBM and Apple.

And, while we're on the subject of costs, it appears that Clive Sinclair may have a slight problem with his Microdrives. Nothing technical . . .it's just that they may not be as close to £50 when they actually appear as was hoped.

THE WORLD INSIDE

For about the last week, our offices were buzzing to the whispered questions like ". . .got a spare ticket?" and ". . .er . .you don't really need two . . .do you?". I must confess to playing dumb (after all I already had mine safely stapled into the lining of my jacket). What, you are probably asking, were these objects that were so much in demand. Well, they were tickets to the Press showing of ***TRON***, Walt Disney's new fantasy film. The appointed evening came and we all duly trudged off to the other end of Oxford Street and emerged, slightly shattered, after some 90 minutes.

The plot, what little there is, is based around the mysterious ENCOM comglomerate who make lots of money out of video games. The trouble is that the boss, Dillinger (David Warner), ripped off the programs from programmer extraordinaire Flynn (Jeff Bridges). Now, as he's been sacked from the company, the only way that Flynn can prove his claim to the software is to break the system via the telephone . . .all good topical stuff this! In order to prevent his accessing the vital file, Dillinger sets up the sinister Master Control Program or MCP for short. This piece of code develops a healthy appetite for other people's software (it even rips off the Pentagon) and starts to close whole chunks of the system down in order to pursue Flynn's relentless search for the missing file.

One of the programmers who finds himself barred from the main computer is Alan Bradley (Bruce Boxleitner) who has been working on a security program called TRON . . .which, as any self respecting Microsofter will tell you, stands for TRace ON. Aided and abetted by Lora (Cindy Morgan), Flynn gains access to a terminal inside the ENCOM building and attempts to enter the master computer. The MCP doesn't like this too much and, by an impressive piece of computer graphics and special effects, reduces Flynn to his component parts and dumps him into the inside of the MCP. Here he finds that the world consists of the alter egos of the programmers and users. The film now turns from reality to pure fantasy and the special effects are stunning, to say the least. Much of the work has been either computer-generated and then enhanced by traditional methods or is pure computer graphics. The plot doesn't really matter from this point on . . .if it mattered at all . . .but there are some marvellous throwaway lines. My favourite sequence is where Flynn's alter ego, Clu, has just polished off one of the MCP's henchmen on the video rink and comes out with the immortal line "It all looked so easy on the other side of the screen". Yes, you've guessed, the insides of the MCP are treated as a giant video game with the goodies and the baddies doing battle on light cycles and the like.

Some of the humour may well go over the heads of non-computer literates, the sequence with the 'bit' being a typical example. Computer related jokes abound, as you might have expected, as the players are made to carry all their vital data around on pseudo-floppy discs which get erased when they've outlived their usefulness to the MCP. Indeed, to mangle one of Ford Prefect's quotes, "A man's really got to know where his floppy disc is".

At the end, as one would expect from a Disney production, wrongs duly get righted — even though there are a large number of holes in the plot which suggest that the cutting room didn't understand the meaning of continuity. The committed film buff might spot rather too many similarities to other Sci-Fi fantasies such as ***Star Wars*** for it to be classed as a great piece of cinematic art, but the mixture of real action, animation and computer graphics will probably make it irresistable to the kids. The film is due for West End release in late October and should be all over the country during late November-early December.

With around $20 million of Disney's money riding on the production (they spent a cool $2 million making a pilot), this will either restore their fortunes after the rather disastrous ***Black Hole*** or add yet another company to the growing list of ailing movie makers. ■

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

COMPUTAMART

AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...AT A GLANCE...

Henry Budgett

CT STANDARDS

Our regular page explaining the meaning of the various symbols we use to make programs portable.

It has been very encouraging to see the number of programs submitted using our standard codes for graphics and other non-printable characters. However, it has also become increasingly clear that some of our readers haven't heard of them and this page is intended to set them out once again.

All standaıds tend to be irksome to adhere to but the ones laid out here are fairly simple and tend to make software easier to maintain by the programmer and simpler to understand for others.

CONTROL THAT CURSOR

Our original standards have now grown with the times. Machines such as the Commodore VIC which have a dual Shift capability can now be incorporated, as can those systems which use Control key functions.

The recently introduced BBC system offers pre-programmed function keys which, we are glad to say, can also be handled by our original coding system. It's nice to see just how well adapted the original standards have become over the last two years! (Indeed, a whole series of books is using them as its *de-facto* standard.) The standards for the cursor controls are given in Fig. 1.

[CLS]	CLear Screen
[HOM]	HOMe cursor
[CL]	Cursor Left
[CR]	Cursor Right
[CU]	Cursor Up
[CD]	Cursor Down
[REV]	REVerse video on
[OFF]	Turn it OFF
[SPC]	SPaCe
[CTL]	ConTroL key
[fn]	Function key (BBC)
[G<]	Graphic left (VIC/MZ-80A)
[G>]	Graphic right (VIC/MZ-80A)

Fig. 1. Our extended set of cursor control standards includes four new functions.

To indicate more than one of the above, an optional number can be placed within the brackets; [4 CL], etc.

The use of square brackets has raised one or two queries. The reason for this choice is that *most* of the common microcomputer BASICs don't use them for specific functions. In fact, at least one machine provides an added bonus by returning a Syntax Error if they are found, a useful check in case you type them in by mistake.

The code [SPC] was added to the list of cursor control codes to get over the problem of indicating just how many spaces are contained in the gap in the printout. The other common variant of the code for spaces is used by the ZX people. Their choice was ⍽ and this crops up in the various newsletters they publish.

The code [RVS] has caused a few headaches. This is really specific to the PET where the character set can be displayed in reversed video. On machines which don't have this facility you should either find a character in the set which is the reversed image of the one you want and use that or simply ignore it and use anything else you fancy! Don't forget, you may have to look up and alter the values used elsewhere in the program.

THE GRAPHIC SOLUTION

It soon became obvious that the techniques applied to the confusing cursor controls could also be applied to the graphics symbols. The following standard is now in general use in programs published in *Computing Today*.

If a graphics character or characters are to be displayed in a listing (as opposed to POKE codes or CHR$( ) codes) then they are indicated by the method shown in Fig. 2.

Several people have asked what the relationship between the POKE value for a character and that of its shifted graphic might be. In general the shifted version of any character will be 64 greater than the value of that character. This applies to both PET and MZ-80K systems in all cases.

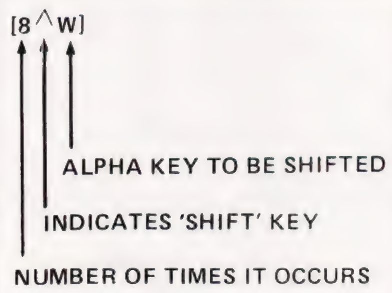


Fig. 2. The way we indicate block graphics on machines like the PET and Sharp. The VIC system of Shift Left and Shift Right is shown in Fig. 1.

This can be taken further to include machines which use a pixel graphics set rather than pre-programmed PET-style characters and the series of codes for these is given in Fig. 3. As is nearly always the case there is one machine to which the standard shown in Fig. 3 does not apply — Tangerine's Microtan/Micron. This machine uses a four by two cell structure for its pixel graphics instead of the Prestel/Teletext three by two cell. The method for calculating the value to assign to 'P' is shown in Fig. 4, and is fortunately nice and simple.

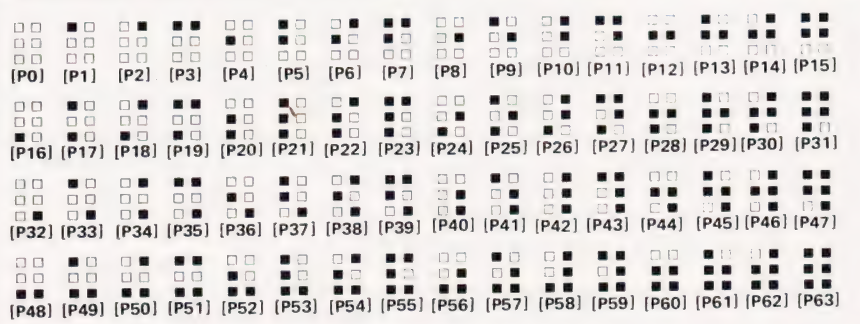


Fig. 3. The standard pixel codes; they will work on most computers which employ this technique as well as for Teletext and Prestel.

1	2
4	8
16	32
64	128

Fig. 4. To convert a Tangerine pixel code into its blocks, simply decode the number into its binary or Hex value and fill in the relevant squares.

MAKING REMARKS

Many people scorn the use of REMs within programs but, during the development at least, they are extremely useful. One of the documentation methods that we use is to keep our back-up copy of our programs on a 300 Baud CUTS tape with all the REMs in place: the working copy, be it on tape or disc, is REMless in order to save space.

It is also good programming 'manners' to give your REMs odd line numbers:

```
3999 REM ** CRASH PROOF INPUT
4000 INPUT "THE NUMBER OF ENTRIES ";A
```

A remarkable number of submitted programs have jumps that go not to the relevant point in the program, but to the REM statement. This can cause severe problems when re-numbering after removing the REMs.

BARCLAYCARD VISA

CLASSIFIED

THE PATH TO SUCCESSFUL SALES!

01-437 1002

Extension 212

ADVERTISEMENT RATES

Semi-Display (min 2 cms)

1-3 insertions **£7.25** per cm
4-11 insertions **£6.75** per cm
12+ insertions **£6.25** per cm

Lineage 30p per word (min 15 words)
Box Nos. **£2.50**

Closing date 2nd Friday of the month preceding publication date.

All advertisements in this section must be prepaid. Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request)

Send your requirements to:

COMPUTING TODAY CLASSIFIED ADVERTISING, 145, CHARING CROSS RD, LONDON WC2H 0EE.

: DISCS FORTH :

A Complete DIY FORTH kit
1) Installation manual **£10**
How to do it + definitions + editor
2) Source code listing for one processor **£10**
6502, 6800, 6809, 8080/Z80, 8086/8088, 9900, 1802
manual + one listing **£19**

B Ready to run
FORTH on disc for most machines incl. PET, TRS80, FLEX, CP/M, IBM, and more **£50**

C Dual 8" disc drives .... **£525 + VAT**
2 × 8" single-sided double-density Shugart drives + box + PSU + intelligent controller

D SAE for details including FORTH books.

MicroProcessor Engineering Ltd
21 Hanley Road Shirley
Southampton SO1 5AP
Tel: Southampton 775482

VIDEO GENIE II + monitor, £70 worth programs. Cost new £510, 10 months old, £350. Lowestoft 0502 86931.

SPECTRUM 3D GRAPHICS 16 or 48K machines — Spectacular plotting under user control of shape, size and perspective — only £6.95. **FIGARO II** powerful numeric database gives you seasonal forecasting, statistics, graphs, data files and much more. Very friendly — no statistics knowledge needed (48K only). £14.95 includes full manual. Access & Barclaycard welcome. Saxon Computing, 3 St. Catherines Drive, Leconfield, Beverley, Humberside.

ATARI 800 32K cassette recorder, Manuals, Programs, £600. Tel: Alan, Orpington 76158 or 01-353 4628.

SOFTWARE IN STOCK

6800	**MONITOR Rom** + Data	**£9.75**
6800	FLT.PT. BASIC Rom + manual	**£14.25**
6800	DISASSEMBLER Listing	**£3.50**
6800	8K BASIC (romable) Listing + manual	**£25.65**
6800	FORTH 6K Tape + manual	**£12.50**
6800	EDITOR ASSEMBLER/DISASSEMBLER Tape + manual	**£22.65**
6800	GAMES BOOK includes Space invaders	**£3.65**
* All 6809 software in position independent code.		
6809	EDITOR ASSEMBLER Rom + manual	**£34.50**
6809	DISASSEMBLER Rom + data	**£11.65**
6809	FORTH Source + manual	**£25.00**
6809	INTEGER BASIC Rom + data	**£12.95**
6809	MONITOR Rom + Source	**£12.65**
Z80	3K BASIC Listing + manual	**£7.95**
Z80	LISP Source listing	**£12.50**
6502	TINY ASSEMBLER Listing + manual	**£5.50**
6502	DISASSEMBLER Listing	**£3.50**
ATOM UTILIROM Rom + data		**£7.95**
ATOM ALPHA-GRAPH ROM + data		**£12.75**

Many more items available send 50p for catalogue. All prices inclusive.

J. MORRISON (MICROS), 2 Glensdale Street, Leeds LS9 9JJ (0532) 480 987

COMPWARE

Popular dot matrix/daisywheel printers at the lowest prices. Special — **NEW MX80 type III only £299.** SAE for full list: 57 Repton Drive, Haslington, Crewe, Cheshire.

COMPU-THINK 400K Disc Drive with controller for 3000 series PT little used £400. Tel: Farnborough (Kent) 52594.

UK101 SOFTWARE ON TAPE

from the guy who wrote "Le Passe-Temps"

GALACTIC HITCHHIKER (8K). An adventure, all in machine code. A beauty! **(£7.00)**
SUPERTREK (8K). Sail boldly through the universe zap-ping moving Klingons in real time. Superb graphics **(£7.00)**
STARTREK (8K). The old favourite, beautifully presented **(£6.00)**
LUNAR LANDER. A real challenge. You won't get down in less than 3 hours. **(£3.00)**
HANGMAN. Excellent graphics. P.E. said so! **(£3.00)**
BASIC TUTOR (8x4K). The only way to learn — at the keyboard. **(£12.00)**
LE PASSE-TEMPS. You NEED this, if you haven't already got it. **(£3.00)**
MAD MONK (8K). It's ready at last! A machine code adventure with some truly remarkable graphics, this programme is in a class by itself **(£9.50)**
These ORIGINAL PROGRAMS are compatible all 2K Monitors and are available for 16x48 and 32x48 displays (including enhanced Superboards)

HARDWARE

These kits are complete in every way:— Fully socketted high quality PCB, all components, switches etc, plus preformed cable assembly for easy interconnection to J1, or our
MOTHERBOARD SYSTEM. Now you can add on all those extras easily. Provides eight, yes EIGHT, fully buffered J1 type sockets. **(£19.50)**
8K STATIC RAM BOARD. (£39.50)
HI-SPEED CASSETTE INTERFACE. At last, a system that works. COMPLETELY RELIABLE 4000 baud (8000 with reasonable cassette) plus software for named file handling. A delight to use. **£19.50)**
VIDEO ENHANCEMENT. Switch selectable 16x48 or 32x48 displays without butchering your computer. **(£19.50).** Monitor EPROMs re-blown to suit for just **£2.50**
8K EPROM BOARD. (£19.50). A 2K Extended Monitor is available in EPROM for **£7.50** and Toolkit in EPROM, **£12.50.**
MONITOR BOARD. Plug into Monitor socket to provide switch selection of up to 4 EPROMS. **(£9.50)**
BASIC 4 Replacement provides a knock-out Cassette Handling System, Old, Dynamic Halt, Single Step and Trace. **(£9.50)**
PRINTERS Phone for latest prices e.g. **EPSON MX80FT II** for **£399.00** incl. Securicor Delivery

Prices include post & packing but excludes V.A.T.
Please write or phone for further details:
MERLIN (MICRO SYSTEMS) LTD.,
93 High Street, Eston, Cleveland. Tel: (0842) 454883.

ACORN ATOM, 12K pools program for Australian/British results. Cassette £7.95. K. Girdlestone, 14 St Hilda Avenue, Barnsley, Yorkshire.

RADIO SHACK Line Printer II for sale. 7 x 7 dot matrix, 80 CPL 115 VAC with 40 pin connector, £200. Lewes 07916 2039.

ACORN ATOM, 12K ROM, 12K RAM, Toolkit, 5V PSU, leads, many programs, £220. Southampton 559792.

B.B.C. DISASSEMBLER on cassette, Model A/B, fully documented, £6.00. Valtos, Newmarket Road, Burwell, Cambridge CB5 0AE.

CRYSTAL RESEARCH LTD

FOR YOUR SHARP MZ80K CP/M 2.21 (XTAL)
BASIC CP/M FACILITIES INCLUDE
• Dynamic file management • General purpose editor
• Fast assembler • Advanced debugging utility

YOUR SHARP CP/M 2.21 (XTAL) PACKAGE INCLUDES
• Hardware modification (if fitted by a SHARP dealer does NOT break the guarantee) • SHARP CP/M 2.21 (latest version) on disc • XTAL Monitor and Operating System • 7 Digital Research manuals • 12 months guarantee and up-dates (on all our products)

CP/M 2.21 (XTAL) FROM £150 + VAT
Ask your SHARP dealer for further details or contact
CRYSTAL RESEARCH LTD CP/M SOFTWARE HOUSES - XTAL CAN HELP YOU ESTABLISH YOUR SOFTWARE ON THE SHARP

XTAL BASIC (SHARP) for MZ80 (A, B& K)

Takes 5K less memory, has all the features of SHARP BASIC PLUS Multi dim strings, error trapping, logical operators, machine code monitor, more flexible peripheral handling, improved screen control, increased list control, auto run, if...then...else - and it doesn't stop there - it grows. You can extend the commands and functions at will - 10K, 12K, 16K, BASIC? SHARP TO XTAL BASIC conversion program is included **£40 plus VAT.**
Bi-directional serial board for your SHARP RS232 compatible <150 Baud to >2400 Baud adjustable. 5,6,7,8 Bit words, plugs into MZ801/0 **£99.50 plus VAT.** Includes software for bi-directional use in XTAL BASIC software for using SHARP BASIC with serial printer and self-diagnostic software for testing Baud rate etc.

Open 0930-1200 1300-1730 except Saturday & Sunday
40 Magdalene Road, Torquay, Devon, England
Tel 0803 22699. 0803 27890
Access and Barclaycard

XTAL BASIC

CASIO FX602 — 20 useful programs only £3.00. T. Harman, 1 Stour Street, Manningtree, Essex.

FORTH

xFORTH — The fastest extensible system used by the professionals.

xFORTH — is true Forth - 79 with full CP/M compatibility and many extensions, including screen editor • from **£48.00**

We also sell other software. For details send S.A.E. On any order please add £3.00 p&p + 15% V.A.T.

SPECIAL OFFER — of 10% off pre-paid orders received before 31st October 1982

AIM RESEARCH
20 Montague Rd., Cambridge CB4 1BX
Tel: 0223 353985

COLOUR VDU FOR MICROTAN

A high performance colour VDU card which plays into the system rack. 25 lines of 64 characters. Teletext character set. Medium resolution graphics. Six colours plus white. Inverse video facility. Supplied with two EPROM's which replace Tanbug and part of BASIC. All previous facilities preserved, including on-screen editing on both screens. Full instruction manual included.

PLUS compatible software: Toolkit — Re-number, Autonumber, graph plotting commands, etc.; Assembler — a comprehensive 6502 editor/assembler, too many features to list; Adventure — a classic style Adventure game, great fun, very difficult. Send SAE for more details, or odrder now for rapid delivery.

INCLUSIVE PRICES: VDU card — version A for RGB monitors only: £69.95 + £1.50 (p&p); version B for RGB monitors or television receivers: £89.95 + £1.50 (p&p). Manual refundable on purchase of card): £2.00 post free. Toolkit (2K EPROM): **£14.95** + 50p (p&p). Assembler (2 x 4K EPROM's): **£29.95** + 50p (p&p). Adventure (23K M/C cassette): **£7.95** + 50p (p&p)

MOUSEPACKET DESIGNS
7 Cedar Close, Grafham, Huntingdon, Cambs PE18 0DZ. Tel: 0480 810550.

SHARP MZ80K ADVENTURE GAMES.
Devils Triangle — Sail to Bermuda but beware! The Bermuda Triangle is where anything can happen — £5.00.
Earth Rescue — Travel to distant galaxys searching for a rare mineral needed for earth survival, but other 'beings' have different ideas! — £5.00.
Games Pack One — Monopoly, Mastermind, U.F.O., Bowling, Muncher — £9.50.
Free Chess with orders over £9.50.
Many more games available. Send S.A.E. for details.
Abacus Software, 20 Rhosleigh Avenue, Sharples, Bolton, Lancs. (Tel: 0204 52726).

DIANETICS: Hardware, the human mind, Software, correctly and incorrectly graded experiences plus unevaluated data. Follow this spectacular journey into the depths and capabilities of the worlds most sophisticated computer — the human mind. Manual written by one of the first Nuclear Physicists, only £4. Dianetic Centre, 3 St. Marys Row, Moseley, Birmingham B13 8HW.

VIC-20 SOFTWARE.... ST1 8 games for the basic VIC £6.50, ST2 5 games for the basic VIC £6.50; for the 6.5k VIC £8.00. And if you want HI-RES games, you NEED graphics Editor at £8.00. Machine code, HI-RES, widescreen ... and more! Soft Toys, 14 Lockharton Avenue, Edinburgh EH14 1AZ.

BBC MICRO

Make the most of your BBC Microcomputer with two books written by the experts. LET YOUR BBC MICRO TEACH YOU TO PROGRAM, by Tim Harntell, **£6.45**, with over 40 complete programs. THE BBC MICRO REVEALED, by Jeremy Ruston. If you've mastered your computer, and want to progress, this is the book for you. **£9.95.**

**Interface Publications, CT,
44-46 Earls Court Road,
LONDON W8 6EJ.**

ADDICTIVE GAMES from Budsoft. Atom cassette, 'Ambush', 'Antitank' (5K text). Vic20 cassette: 'Sniper', 'Mines'. £4.00 each. Cheque or P.O. to J. A. Hudson, 6 Penns Wood, Farnborough, Hants.

RAMKRAM Atom Software. Munchie Man, Wrap Trap, Death Ray, Massacre, Killers and many more. £3.75 each. SAE for details, Ramkram, 118 Burford, Brookside, Telford, Shropshire.

PRINTER TELETYPE KSR 33, RS232 I/F. £66.66. Computer keyboard plus case 66 + n quality keys £6.66. Tel: 034466894.

nascom

Compression Assembler NEW **£12**
The best of its kind for the Nascom, this full Z80 assembler compresses source to half the normal size. Thus bigger programs fit in memory & loading/saving is twice as fast. As a bonus it assembles 3000 1pm! It is 6K and includes a ZEAP conversion utility.

Extension Basic 200 sold **£15** (£25 in ROM)
Adds these keywords to ROM BASIC: IF . . . ELSE, REPEAT . . UNTIL, WHILE . . WEND, AUTO, BREAK, CALL, CHECK, COPY, DEC, DELAY, DELETE, EDIT, FIND, GET, HEX, INKEY, INLIN, LINE, PLOT, PRINT , PUT, RENUMBER, REDUCE, SPEED, TEST, TRACE, VDU, WRAP, XLIST & XREF. And you can add more yourself. EB is only 4K.

Adventure Quest: see our ¼ page ad 16K=**£8**, 32K=**£10**
Colossal Adventure: see ¼ page ad 16K=**£8**, 32K=**£10**
Asteroids: fast action in space. Must be played! **£8**
Bomber: flatten your least favorite cities **£5**
Missile Defence: fast action nuclear defence **£8**
Nightmare PORK: battle the pigs and the puns **£5**

FREE P&P. NO VAT. Money back if not happy. Fully documented. (Nascom 1s need Nas-Sys/Cottis B.)
Send order, or request for full details to:

LEVEL 9 COMPUTING
229 Hughenden Road, High Wycombe, Bucks HP13 5PG.

ACORN ATOM 12K RAM 12K ROM, (F. point and graphics) 4K Utility ROM. Manuals plus Software, £145. 0723 77378.

ZX USERS

Join the National ZX Users' Club, and make the most of your ZX81, Spectrum or ZX80. Each month we publish the magazine INTERFACE, which contains a host of programs for each machine, as well as members-only offers, hints, tips, contact addresses and more. We'll also help you with your hardware or programming problems. Join the country's largest users' club. £1 gets you a sample copy of INTERFACE and full club details.

**National ZX Users Club, CT,
44-46 Earls Court Road,
LONDON W8 6EJ.**

BBC PADDLES WITH FREE ASTEROID BELT GAME — £11.50. Computer to Cassette leads (Type 1), 5 pin din. (Type 2) two 3.5mm plugs and one 2.5mm plug. (Type 3) 5 pin din and remote. (Micro comes with incomplete lead) — all £4.50. Electronics applied, 4 Dromore Road, Carrickfergus, Co Antrim BT38 7PJ.

32K PET 2001, new ROM, over 50 programs, many other extras. £410 ono. 544 9548.

VIC 20

It is essential to protect your Microcomputer especially from DUST which can affect the keyboard. Our specially designed fitted covers are made from high quality reinforced vinyl with cut-outs for wiring etc.

DUST COVERS

Send **£3.45** (inc. p&p) to
Earlcape Ltd, Lowfields Rd, Leeds LS12 6BT.
State VIC or **BEEB**

BBC MICRO

SPECIAL OFFER

16 Bit
TEXAS T199/4A

FOR ONLY £179 LIMITED PERIOD
(£209 inc. VAT & carriage)

COMPASS LTD Tel: 0257 42652
6 Market Street, Standish, Lancs.

BBC MICRO. The Artful Dodger. Cunning Graphics game with sound. Cassette £4. Brownson, 20 Upland Court Road, Harold Wood, Essex.

PET UPGRADE

We can upgrade your large keyboard PET at a fraction of the "New Price" difference.

8K to 16K £44.00
16K to 32K .... from £56.00
8K to 32K £69.00
If expansion area drilled with quarter inch holes .... £10.00 extra
All new RAMs fitted sockets.
Fast, 2 hour while-you-wait service.
Tel: Mick Bignell of Miroserve on 01-953 8385.

ACORN ATOM, 12K + 12K, PSU Manual, plus lots of software, £150 ono. (Sheffield) 0742 650695.

The Polytechnic of North London

EXPERT SYSTEMS

**Polytechnic of North London
(Monday, 20 December 1982)**

An intensive 1-day course with a practical emphasis plus 'live' demonstrations.
Knowledge-based Expert Systems are at the heart of the Japanese 'Fifth Generation' computer project. The deadline for their Inference Engine is 1990 Come along and find out what is on the horizon
Course fee of **£40** includes lunch & drinks Write to:
Richard Forsyth, Dept. of Mathematics, Statistics and Computing, Polytechnic of North London, London N7 8DB.

ACORN ATOM 12K RAM + 12K ROM, 2 F.P. ROM. All leads and power adaptor, Manual book plus 3 Acornsoft tapes. Excellent condition. £190. Dudley 58259.

48K SHARP MZ80K with four months guarantee. Software worth £40 including Data Base Games. All as new. £325. Ring Leicester 864945 evenings only.

TRS-80 L2/GENIE SOFTWARE (16K)

Cassette 1: Space Invaders (m/c), Mastermind (finds your code in seconds), Apollo (avoid the aliens), Machine code copier, Defuse and Rate.
ALL SIX PROGRAMS ON ONE CASSETTE FOR ONLY
£4.75

Cassette 2: Space Defender: A fast, professional quality m/c program in graphics with mystery ships, top 10 ladder continuous display of Score etc. A fast, real time action packed game.
Apollo 7: An exciting m/c space game with sound effects and graphics.
BOTH PROGRAMS ON ONE CASSETTE FOR ONLY
£4.75.

Cassette 3: SPEAK: Turn your TRS-80 into a speak and spell machine with this m/c program. No extra hardware required. MATHS: Questions & answers on +, −, × & ÷.

Send SAE for details. Each cassette costs **£4.75** (or 3 for £12) from: **K. Meeran, 32 Lismore Road, S. Croydon, Surrey CR2 7QA.**

ZX81 CASSETTES

DRAWING BOARD (16K) £4.20
M/C aided. Pictures can be stored, recalled, merged, scrolled 4 ways and put into a PRINT statement for use in your programs

"BOARD" GAMES £3.20
(For 2 players. 16K). Snakes & Ladders. Othello. Pipeline. Four in a Row

1K GAMES PACK £3.95
10 well written games — Flyswat. Guess the Word Simon. Fruit machine. Duckshoot. Lander. Find the Sub etc.

DATABASE (16K) £4.95
Data storage and retrieval — Side 2 with example — preserved steam locos. **Post Free**

CADSOFT
24 St James St., Cheltenham. Glos GL52 2SH

ATOM OWNERS! 6K, FP. Advanced scientific calculator. Handles complex numbers, binomial expansions, integration etc. Detailed documentation, applications. Cassette £5.00. Tahir Yaqoob, 11 Alexandra Road, CR0 6EY.

NASCOM II 48K graphics, Zeap, NasDis, Debug, Toolkit, NasSys, NasSys 3, Sargon, Ay-3-8910 sound board, boxed with documents. £300. Tel: Lancaster (0542) 36592.

ZX81 (16K) Satellite Guardian. Amazing fast graphics: Earth actually rotates in full view. Can you defend the cities of Earth? £2.95. Atom (4K) Supertrack — Fast moving road-race. The Atom assesses your driving skills. Superzombie Munch or be munched. Both for £2.95. J. Owens, 3 Norfolk Road, Uxbridge, Middlesex.

DON'T LOOK
Anywhere else for lower EPSON or OKI printer prices. Send S.A.E. for list.

COMPWARE
57 REPTON DRIVE HASLINGTON, CREWE, CHESHIRE CW1 1SA.

NASCOM 2, 16K. Vero-Rack, Keyboard case, NAS-SYS 3. Also Zeap (ROM), Unused kit IO High-Res Graphics, NAS-GRA, Sargon Chess. Offers £325. Hemel Hempstead 47596.

BUTTERCRAFT SOFTWARE for TRS-80 and Video Genie.
Auto-Graphics-82 for easiest possible static, moving and animated graphics. £8.95. (". . . best documented program I've ever bought, and no bugs." C. L., Devon).
Auto-Sonics for easiest possible sound effects. £6.95. (". . . still quivering with excitement after using the best sound effects program I have ever seen!" R. T. Netherlands).
The Brood are fast, intelligent and dangerous! The price £5.99. ("... certainly the best pac-man for TRS-80 . . . very addictive." F. R. Leeds).
Accel/Tsave turns your S-L-O-W Basic program into machine code that runs up to fifty times faster. £24.90. Buttercraft Software, 14 Western Ave-. Riddlesden, Keighley, Yorkshire.

32K NASCOM ADVENTURE

The REAL Adventure begins here, with this full-spec 32K version of the original classic mainframe Adventure at last available for your Nascom in fast efficient Z80 code. No discs, no graphics needed — just 32K RAM and NAS-SYS.

Somewhere nearby is Colossal Cave, where others have found fortunes in treasure, though some who enter are never seen again. . . Explore a vast underground labyrinth of caves, twisty passages and exotic rooms; outwit fierce enemies; collect all the many treasures and become a Grand Master!

Using the most advanced text compression techniques the authentic mainframe messages have been squeezed into 32K.

Supplied on high-quality C60 cassettes with full instructions. Only **£20** or a large gold nugget!! Add 55p post and packing. Discount for Adve 16K owners. Send cheque or PO stating tape format (N1, 300), or SAE for further details on this and other software, to:

SYRTIS SOFTWARE
23 Quantock Road, Bridgwater, Som.

ACORN ATOM power supply 5volt/ amp £46.00. EPROMS PROGRAMMED from £7.50. EPROM BOARD plugs into Acorn bus from £10.00. UTILITIES on cassette from £3.50, on EPROM from £8.00. Send 20p in stamps for further details. ATOMIC COMPUTER SYSTEMS, 1 COWLEY CRESCENT, PADIHAM, BURNLEY, LANCS BB12 8SX.

OSI/UK101 PROGRAMS

Details of all known programs available in the UK for all formats and all monitors.
£1.00 only
From:
TRUST SOFTWARE SERVICES
3 Spring Bank Lane,
Rochdale, Lancs OL11 5SE.
If you are selling programs and haven'y yet heard from us, please drop us a line.

OSI/UK101 PROGRAMS

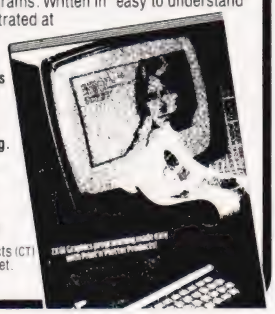
SHARP MZ80K SOFTWARE. Gambling games (Blackjack, Craps, Undercover, Addup, Hi-Lo). All £8. Battleships £4. For 24K+ Machines. J. Porterfield, 805 Frobisher, Dolphin Square, London SW1. 01-834 3800 ext. 805 Frobisher.

DISASSEMBLED PRINTOUT from single/three rail Eproms. 8080/85, 6502, Z80 mnemonics, £1/K. State CPU. Programming your Eprom from listing £2.50/K. My Eprom add £2.50. Small mods 10p + 1p/byte. UK101 BASIC 1 modified (proper error reports), BASIC 3 modified (handles dimensioned string arrays); £6 each. Erasing free with programming, otherwise 10p/chip. P&P 30p, free for orders over £6. Test/repair/modify Triton, UK101, Nascom kits. Other services considered. Dunstable (0582) 600913.

"ZX GRAPHICS PROGRAMMING MADE EASY"

Full Colour A4 Manual, 24 pages packed with ZX Graphics Programming techniques and ideas for games and 'serious' Programs. Written in 'easy to understand' language and illustrated at every stage.

Ideas include:
Information Graphics
Sketch Pads.
Saving your 'Art'.
Making serious programs interesting.
Graphics Stringing.
ZX Printer Graphics



£1·50
(INCLUDING U.K. POSTAGE)

Print 'n' Plotter Products (CT)
19 Borough High Street
London SE1 9SE.

UK101 LAMPOON. Two player game, £5.50. Outpoint your partner or catch the Lampoonite to win. Good graphics. Cegmon. Formats available: Standard, Basic 4, Basic 5, any combination. Please state preference. Cheques/PO's to T. Cain, 19 Central Road, Ramsgate, Kent.

ACORN ATOM, £200, 12K ROM, 12K RAM colour board, plus printer interface. Kidderminster (0562) 741631 after 6 pm.

NASCOM 2 PSU 16K RAM portable TV 8K Basic Tool-Kit Zeap-ASM, DIS-ASM FDC PCB Manuals, £260. Tel: 01-221 6864. 2 Westbourne Crescent, London W2.

SPECTRUM, VENTURE. Exciting new game for the Spectrum (7 games in 1). In colour, with sound & fantastic screen effects. 16K & 48K version supplied on one cassette for £6. Bobker, 29 Chadderton Drive, Unsworth, Bury, Lancs.

VIC-20 PONTOON. Win £££'s from your friends. Full colour graphics on unexpanded 3½K. Cassette £3.95, Listing £1.45. Also available for PET. Phone 0782 413351.

COMPUKIT JOYSTICKS

New Compukit owners can fit Atari joysticks to their computers with our complete kits — no more straining at the keyboard trying to remember which key to press. Easy to use in your own programs, full details supplied. Kits include joystick(s) and full instructions. **PRICES £12.95** single **£23.95** double. Program modifications for all our games also supplied.

NEW UFO — Blast the UFOs as they speed towards you in the rock matrix
SURROUND — Our version of the popular game. (With over 20 speed levels). These games **£3** each
SPECIAL OFFER — Only **£2** each if bought with a joystick.

Asteroids, Space Fighter, Invaders, Astro Sled, Chaser:
LAST MONTH of SPECIAL OFFER .... **£1.95** each

NEW VIC-20 SOFTWARE

Astro Sled — Fly through the jagged space canyon. With colour and sound. Fast machine code routines **£5**
Machine code monitor — BASIC monitor for entering and experimenting with machine code **£2**
Graphics aid — For developing and using programmable graphics characters **£2**

ALL FOR UNEXPANDED VIC-20

P&P FREE on orders over £5, otherwise 50p

Arcadia Software
4 Sunningdale Avenue, Swansea SA3 5HP.

PROGRAMMERS WORKSHEETS. Sample pack £1.10p State Basic or Machine Code and machine, or SAE for list. BUSINESS SERVICES(P1), 16 Grosvenor West, Baldock, Herts. SG7 6NZ.

SHARP PC 1211 Computer, £65. CE 122 Printer, £60. CE 121 Cassette Interface, £10. Hurry, few left. Add £2.50 p&p. S. O'Connor, 6 Pinewood, 21 Oaklands Road, Bromley, Kent.

SUPERBOARD II with 16K RAM and Screen Enhancement Kit (10 screen formats). Max 64 x 25 plus Currah Stringy Floppy plus Cegmon plus Converted P.T.V. for use as monitor, plus metal case, plus P.S.U. £325 ono. Tel: 061 643 2578 after 7 pm.

PET PROGRAMS, old Rom 2001 Space-Invaders, Radio Amateur Package, Maths Package, Fruit-Machine. £6 for 1. £10 for 2. £14 for 3. £16 for 4. G. M. Tibbert, Fir-View, Lyndon, Oakham, Leics LE15 8TU.

32K NEW ROM PET, Cassette, Soundbox, Toolkit, much software including Arcade games and Adventures, and approximately fifteen editions of "Cursor". £500. Telephone Cuffley 2865.

SHARP MZ-80K 48K, Sharp & Xtal Basics, FIG-Forth; I/O Unit with Universal Card; lots of games, Dissassembler, etc; Service Manual; User group literature, £390. David, Tel: Egremont (0946) 820436 evenings.

VIC-20 MICRO

Features similar to Popular PUB machine. Semi controllable NUDGES up to UNLIMITED using HOLD or GAMBLE. Reels NUDGE up & down. SOUND EFFECTS. GREAT GRAPHICS. Requires min 8K RAM expansion. Price £5.50. Cheque or PO to: **J. Ingham, 72 Ardwick Street, Burnley BB10 1BJ.**

CAN YOU WRITE GOOD PROGRAMS? Do you want to be paid for what you enjoy doing? A reputable software house is interested in hearing from YOU. Why not write now to: Box No. CT 2029, A.S.P. 145 Charing Cross Road, WC2.

VIC 20 GAMES. Maze Chaser action game or Strategic Connect 4. £3.50 each or both for £6.00. J. Owens, 189 Bewley Drive, Liverpool L32 9PF.

SUPERTEXT ZX81 — 16K

Print any alpha/numeric character
10 TIMES NORMAL SIZE

Fill the screen with 5 lines of 10 characters. Full cursor position — space — rubout — prints + - × ÷ =. Ideal for displays, classroom demos., etc.

Or, if you own a printer, make your own mini-posters — labels — titles.

Cassette £4.95 (inc. p&p) from
— COMTEXT —
24 Honeybottom Rd., Tadley, Hants RG26 6JP.

Artic Computing

ARTIC COMPUTING are looking to buy high quality programs for the ZX81, ZX-Spectrum and BBC Computer, to be included in our future catalogue. Any type of program considered from Games to Business Applications. Please send copy of program, preferably on cassette, with instructions and other information to:
Richard Turner, ARTIC COMPUTING LTD., 396, James Reckitt Avenue, Hull, N. Humberside HU8 0JA.

SPECTRUM PROGRAMS WANTED. Earn 40% royalties. Send software on cassette plus details to:— Popular Software, 15 Nottingham Road, Barrow-upon-Soar, Leices LE12 8HZ.

VIC 20 with Cassette. M/Code, S/Expander and Invaders cartridges worth £330. Accept £200 ono. Phone Leatherhead 379349 evenings.

PARAPHYSICS JOURNAL (Russian translations); Psychotronic Generators, Kirlianography, gravity lasers, telekinesis. Details: SAE 4 × 9". Paralab Downton, Wilts.

ZX SPECTRUM PROGRAMS. Diary, Personal Budget, Bank Account, Address/ Telephone/Birthday file. All four on tape for £4.95. Sussex Software, Wallsend House, Pevensey Bay, Sussex.

SHARP MZ80K SOFTWARE. Cassettes for sale in one lot (changed to disc). Over 30, mostly games, plus Teach Yourself Basic in Ten Lessons, C.A.L. Physics, and Address Book Filing. £80 ono. Send S.A.E. for list. B. Franks, 5 Torkington Road, Wilmslow, Cheshire.

T199/4A SOFTWARE on tape, from £1.95. Send s.a.e. for list. Apex Trading Ltd., (CT), 115 Crescent Drive South, Brighton BN2 6SB.

TANGERINE OWNERS, 80K Romswitch board, provides software switching of Basic Rom space, also battery backed 40K Static Ram board. Telephone evenings, Richard (Romswitch) 0602 267739, Trevor (Ram) 0602 392965 or SAE to 54 Westerlands, Stapleford, Nottingham.

ZX-81 32K RAM, £100 o.n.o. Also software and magazines, owner upgrading system. More details 01-789 0193.

SINCLAIR SPECTRUM, Machine code 'Break-out' for 16kb model. £3.50. Cheque /P.O. to J. Lewis, 59 The Chase, London SW16 3AE.

FIND-A-FRIEND through FIND-A-FRIEND's new confidential, inexpensive service. Your ideal friendship/relationship - all ages - countrywide. SAE/Telephone: FIND-A-FRIEND (CMP), Temple House, 43-48 New Street, Birmingham B2 4LH. 021-429 6346.

PLEASE MENTION COMPUTING TODAY WHEN REPLYING TO ADVERTISERS

AD INDEX

COMPUTING TODAY — CLASSIFIED ADVERTISEMENT — ORDER FORM

1.	2.	3.
4.	5.	6.
7.	8.	9.
10.	11.	12.
13.	14.	15.

Advertise nationally in these columns to over 100,000 readers for only **30p** per word (minimum charge 15 words). Simply print your message in the coupon and send with your cheque or postal order made payable to Argus Specialist Publications Ltd to:

CLASSIFIED DEPARTMENT COMPUTING TODAY
145 Charing Cross Rd, London WC2H 0EE.
Tel 01-437 1002

Name . . . .

Address . . . .

. . . .

. . . .

Tel.No.(Day) . . . .

Please place my advert in Computing Today for ☐ months. Please indicate number of insertions required.

You can't get a Home Computer from Texas Instruments under 16 K RAM.

Make the right move into computing with the Home Computer from Texas Instruments. It gives you a large combined RAM/ROM capacity up to 110 K Byte and the ability to expand with a full range of peripherals and software. So as your knowledge of computers increases the TI Home Computer will grow with you.

Just compare the versatility of the TI Home Computer with its price – you'll find it real value for money that will prove to be a good long term investment.

The TI-99/4A is a sophisticated computer designed not only for the beginner with its ease of operation, but also for the professional with its vast computing power through a 16 bit microprocessor. And it simply plugs into an ordinary household TV set.

With its high resolution graphics with 32 characters over 24 lines in 16 colours (256 x 192 dots), 3 tones in five octaves plus noise, and BASIC as standard equipment and options such as other programming languages – UCSD-PASCAL, TI-LOGO and ASSEMBLER – and speech synthesis, you'll find that the TI 99/4A more than compares with the competition. Especially when the starting price is around £200. When you want to solve problems there are over 600 software programs available worldwide – including more than 40 on easy-to-use Solid State Software® Modules.

After all, from the inventors of the microprocessor, integrated circuit and microcomputer, it's only natural to expect high technology at a realistic price.

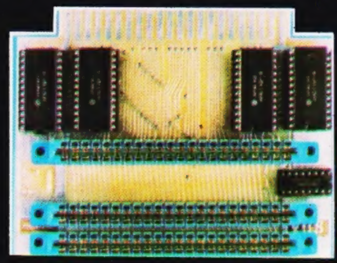
UNBEATABLE VALUE AT AROUND £200

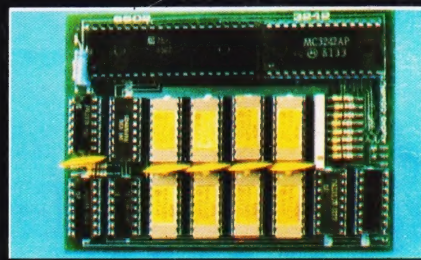
We'll help you do better.

TEXAS INSTRUMENTS
LIMITED

MORE MEMORY FOR MICROS!

look at what we can offer:



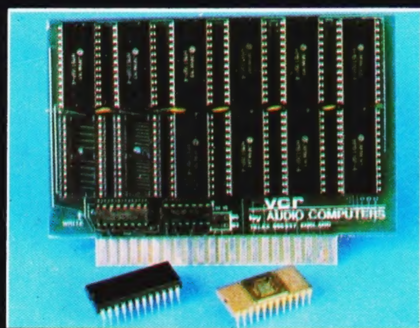


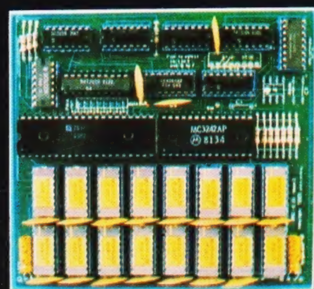
RAMPACKS
for SINCLAIR ZX81

VCS 8K for VIC
8k RAM+3 slots £44

DU01 for ATOM
64k RAM only £70

PARALLEL PRINTER
INTERFACE
AND 32K
INTERNAL MEMORY
AVAILABLE
FOR SPECTRUM
NOW!

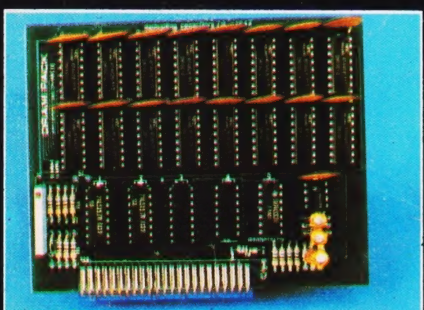


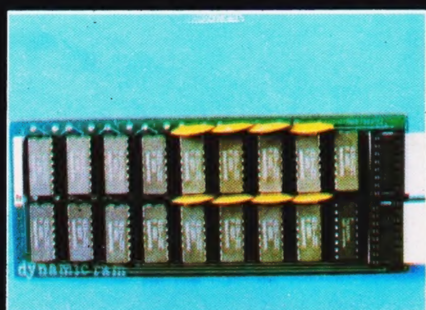


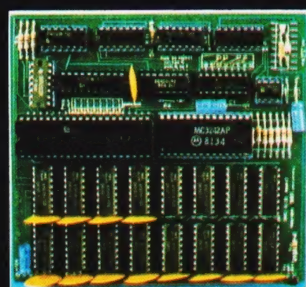
add up to 20k RAM+16k ROM

VCR 20 for VIC
4K-£24, 2k increment-£5

DRC for PET
64K-£80 128K-£130







32K bytes - £29

TRS80-V.GENIE
32K add on RAM £46

UK101:
32K add on RAM £69

FOR A FREE BROCHURE, RING LINDA OR SUE ON SOUTHEND (0702) 613081
FOR CREDIT CARD ORDERS, RING JACKIE OR PAM ON SOUTHEND (0702) 618144

CHEQUES AND P.O.'s TO:

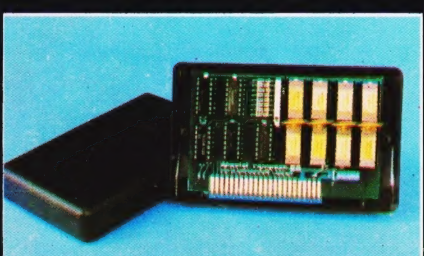
AUDIO-COMPUTERS

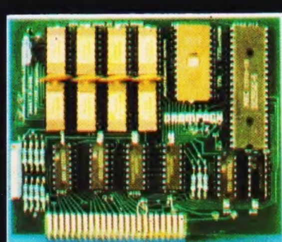
87 BOURNEMOUTH PARK ROAD,
SOUTHEND ON SEA – ESSEX SS5 2JJ

ALL PRICES INCLUDE V.A.T. AND POSTAGE

TELEX 995337 G AUDCOM

A TRADEMARK OF SOLIDISK LTD.





56K – exceptional low power consumption - £44.90

64K + Eprom programmer & user port – only £79

WE CAN ADAPT THE ABOVE PRODUCTS FOR YOUR NEW SPECTRUM FOR JUST £5 EXTRA